Architecting for Artificial Intelligence with Emerging Nanotechnology

Sourabh Kulkarni[†], Sachin Bhat[†] and Csaba Andras Moritz

Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA, USA, skulkarni@umass.edu, sachinbalach@umass.edu, andras@ecs.umass.edu

[†] These authors contributed equally to this work.

ABSTRACT

Artificial Intelligence is becoming ubiquitous in products and services that we use daily. Although the domain of AI has seen substantial improvements over recent years, its effectiveness is limited by the capabilities of current computing technology. Recently, there have been several architectural innovations for AI using emerging nanotechnology. These architectures implement mathematical computations of AI with circuits that utilize physical behavior of nanodevices purpose-built for such computations. This approach leads to a much greater efficiency vs. software algorithms running on von-Neumann processors or CMOS architectures which emulate the operations with transistor circuits. In this paper, we provide a comprehensive survey of these architectural directions and categorize them based on their contributions. Furthermore, we discuss the potential offered by these directions with real world examples. We also discuss major challenges and opportunities in this field.

CCS CONCEPTS

• Hardware~Emerging technologies • Computing methodologies~Artificial intelligence

KEYWORDS

Artificial Intelligence, Bayesian Networks, Computer Architecture, Emerging technology, Nanodevices, Nanoscale Architectures, Neural Networks, Neuromorphic Computing, Probabilistic Graphical Models.

1 Introduction

Artificial Intelligence (AI) has shown great promise in an ever-increasing number of applications such as self-driving cars[1][2][3], natural language processing[4][5][6], computer vision[7][8], personalized medicine [9][10] and many more.

There are a variety of AI models currently deployed in the real-world applications. These models are inspired by progress in number of domains such as neuroscience, calculus, probability theory and statistical analysis. Currently, the most prevalent approaches of AI used in practice are Neural Network models (NNMs) and Probabilistic Graphical Models (PGMs). Real-life applications set a lot of constraints on AI models. For example, genetic networks require the use of billions of parameters to model genetic information. Furthermore, self-driving cars are required to take decisions in real-time which puts pressure on the performance they need to reach. In addition to this, cybersecurity and other applications would benefit from learning in real-time. Emergence of IOT devices and appliances push for the development of very low-cost, power- efficient solutions. The breakthrough in AI has been enabled not only by the advancement in algorithms but also by the use of tensor-compute based software systems accelerated by GPUs [11]. However, even systems powered by GPUs, run into resource constraints [6] and require several weeks for learning while consuming large amounts of power[16]. More recently, custom hardware solutions such as based on FPGAs [15][16][17] and ASICs [18][19][20][21][22][23] have been designed. FPGA and ASIC-based implementations are not as versatile as software approaches; however, they are tailored to achieve best results for a given model or a few models at the most. When compared with GPUs, FPGA-based implementations are up to an order of magnitude better in energy efficiency and marginally better in performance-per-watt [16]. Similarly, ASIC-based approaches achieve up to two orders of magnitude better power efficiency and performance [18] vs. GPU based systems. As we shall discover in this survey, even though these systems indicate great progress, a lot of scope remains for improvements on all key facets, beyond GPU/FPGA/ASIC directions.

Conventional approaches for AI models, even the hardware-based directions, are inefficient because they rely on several layers of abstraction. In comparison, new directions with emerging technology can often bypass these layers by directly implementing the conceptual computational frameworks of AI in the physical layer. We refer to these directions as 'Emerging Nanotechnology-Enabled AI'(ENAI). There are a variety of nanodevices that provide new capabilities towards AI. These include in-memory computing enabled by unique related computational behavior [24] as well as implementing neurobiological functionalities [25]. Examples of specific useful nanodevice capabilities include fast and low energy switching between multiple analog states [26], persistent storage, inherent stochasticity, oscillatory behavior, and directly implementing Hebbian learning. At circuit and architectural levels, research is geared towards realizing AI models using new emerging technologies without layers of abstractions directly emphasizing the underlying device principles.

There have been several survey papers which review use of emerging technology for neuromorphic and neural network architectures and frameworks [27][28][29]. There has not been, to the best of our knowledge, works which encapsulates the broader field of AI which not only includes neural network and related models but also statistical and probabilistic graphical models. Furthermore, these surveys are typically limited in scope for the devices they cover. This survey aims to be broader in both aspects, where the wider scope of AI is captured along with a broader gamut of nanodevices.

In this paper, we identify four key ENAI related directions: i) Circuit-level-focused works not yet reaching architectural scale; ii) Architectures *combining* CMOS technology with nanodevice-enabled unique functionality; iii) Novel architectures that utilize multiple/diverse nanodevice capabilities to achieve ENAI with *minimal* CMOS support; iv) Related integrated circuit technologies to efficiently *realize* aforementioned directions (e.g., the same way as CMOS and associated material stack enabled the large-volume production of digital systems in the past). These approaches are described briefly below and will be discussed in detail in later sections.

Key Computational Circuits for ENAI: Prior to designing complete architectures for ENAI, key computational circuit blocks need to be identified and designed. For example, in NNMs, major computational blocks include synaptic weight and neuron circuits whereas in case of PGMs, it could be conditional probability tables and belief update units etc. These works may entirely rely on computer simulations, vs. actual prototyping, using device models or, in some cases, even fabricate actual circuits. Since the focus is on demonstrations of key functionalities, these latter directions usually rely on computer-aided test equipment and software for signal conditioning, I-V characterization, testing, process monitoring, analysis of results etc.

Nanodevice-aware Architectures for ENAI: This category encompasses works that design complete architectures for ENAI by augmenting CMOS technology with emerging nanotechnology. This involves often building on the contributions by the works in previous category. These works may simply assume that some CMOS integration is available. For example, magnetic devices and memristors may be compatible at the material-system level with current CMOS manufacturing, although the integration in efficient ways is still an open question. This integration is in fact the research target for the fourth category of papers.

Toward All-nanodevice ENAI Architectures: Designing computational circuits for AI involves substantial complexity. Circuit design efforts in this category aim to collapse this complexity using *mostly* emerging nanodevices. This involves engineering new nanodevice properties that could be directly utilized for AI computations. Devices exhibiting spiking behavior for neuronal dynamics, plasticity for Hebbian learning, stochasticity for encoding probability distributions are some of the directions pursued.

Integrated Circuit Technology for ENAI: Research focusing on transformative ways to provide an integrated solution for all technology aspects (device, structural features, materials, circuit styles/components) specifically designed for AI. These technologies have features that are architected to solve issues such as connectivity, manufacturability, 3D integration, material stacks, integration with CMOS subsystems, etc. Collectively these are referred to as ‘ENAI fabrics’.

1.1 Structure of the Survey

The paper is organized as follows: Section II provides a brief overview of the prevalent AI models as well as various nanodevices employed; Sections III-V covers the three major approaches in ENAI and discusses most representative research efforts; Section VI covers the emerging ENAI fabrics development efforts; Section VII includes comparative analysis, additional discussions and conclusion.

2 Background

The domain of ENAI extends into various other domains of expertise and its discussions will involve various terminologies. In a broad sense, ENAI approaches enable the key AI models, and utilize key device and circuit properties in doing so. This section provides some preliminary background knowledge on AI models and nanodevices to facilitate discussions in later sections.

2.1 AI Models

The ever-increasing success of AI is, to a great extent, related to the development of various AI models. These models and their mathematical frameworks have been worked upon over the last century and they continue to be worked upon today. The two major functions associated with AI models is learning them and performing inference on them. Learning, also called training, is a process of optimizing model parameters from data while inference refers to using *learned* models to predict missing values or future outcomes from new observations. Throughout this survey, training and learning will be used interchangeably. We briefly discuss the prevalent AI models, a majority of which can be categorized into neural network models and probabilistic graphical models, which the paper focuses on (see Figure 1).

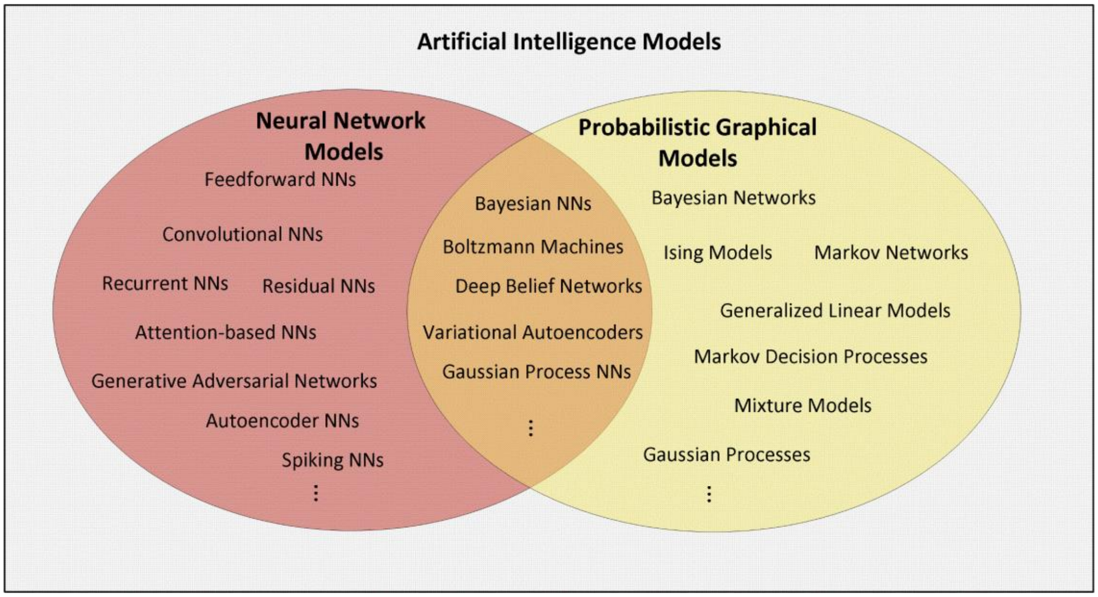

Fig. 1. The ‘Map’ of AI Models. Majority of AI models can be categorized as either NNMs, PGMs or hybrids thereof. This survey shall focus on architectures that are based on these models. While many within the AI community consider NNs and PGMs to be subfields of Machine Learning (ML), which by itself is a subfield of AI; here we place them directly under AI as these aforementioned distinctions are becoming quite blurry these days. While there are AI models outside these two general paradigms, there is not much work in ENAI regarding them.

2.1.1 Neural Network Models (NNMs)

NNM is an AI computing framework that can learn to perform classification and clustering tasks by learning features of the data. At the core, NNMs consist of large number of simple processing units called as neurons that are densely interconnected in layers. The information coming into these neurons are weighted (multiplied) by synaptic weights. Multiply-accumulate operation between the inputs and synaptic weights is the most dominant computation in the NNM graph. Networks which have a distinct training and inference phase are called static networks while networks that continue to evolve during the inference phase are dynamical networks. Significant part of the training phase is used for evolving the synaptic weights in NNMs while rest of the time is used for optimizing the so-called hyper parameters such as number of layers, learning rate, number of training epochs etc. Deep learning (DL) refers to NNs with large number of layers.

NNs can be classified based on neuron functionality, connectivity type, learning algorithms, type of signal integration, applications etc. In multi-layer feedforward networks, the neurons and synapses are connected in a strictly forward fashion whereas recurrent networks have feedback loops. Convolutional neural networks (CNNs) are deep, feedforward neural networks with convolutional filters that are specially tailored for computer vision applications [30]. Recurrent neural networks (RNNs) have feedback connections which allows for a temporal dynamic behavior. Another major class of neural networks called Spiking Neural Networks (SNNs) use spikes (all or nothing signals) unlike DNNs which output real numbers. The information is encoded into timing and frequency of spikes of neurons. Some SNNs use bio-realistic models of neurons. These systems are strictly dynamic in nature since learning is integral part of these models. Generative models like Autoencoders and Deep Belief Networks (DBNs) model joint probability distributions of inputs and outputs to extract deep hierarchical representation of the data. Generative Adversarial Networks (GANs) consist of two neural networks contesting to fool each other.

The two major learning paradigms for NNMs are *Supervised* and *Unsupervised* learning. Supervised learning is a learning paradigm that maps inputs to outputs based on the example input-output pairs of training data. Backpropagation is the mostly commonly used supervised learning algorithm for NNs in which synaptic weights are incrementally updated by calculating the gradients with respect to a loss function [31]. Unsupervised learning algorithms, which are mostly local in nature, find underlying structure of data and are useful when the data is unlabeled. Some examples of unsupervised learning algorithms are Hebbian learning [32] and its variants like competitive learning [33], Spike-Timing Dependent Plasticity (STDP) [34] etc. CNNs are tailored for computer vision applications while RNNs are used for signal processing/speech recognition, captioning systems, etc. SNNs have been used in various vision applications, but they are most popular in neuroscience applications. Autoencoders and DBNs are used for generative learning, image processing/denoising and generating new images.

2.1.2 Probabilistic Graphical Models (PGMs)

PGMs are graph-based representations which encode joint probability distributions over a set of random variables. The random variables are represented as nodes and relationships between them are represented by edges. Directed edges may encode causal relationships, while undirected edges represent non-causal dependencies. The graph is a compact representation of the joint probability distribution among the random variables. PGMs can be broadly classified into two types based on whether the edges of the graph are directed or undirected. Directed graphs consists of Bayesian Networks (BNs) while undirected graphs consist of Markov Networks (also known as Markov Random Fields or MRFs) and Boltzmann Machines. This allows these models to capture fundamentally different relationships between variables and hence used in modelling different phenomena. While being representationally different, they share the probability arithmetic involved in the inference and learning process.

Learning of PGMs has two aspects - learning the structure of the graph and learning the parameters. Learning the structure of PGMs in NP-Complete [36] and is usually done via search-optimization techniques that minimize an objective function, typically KL divergence [37] or Evidence Lower Bound [35] and is an area of active research. Learning parameters is NP-hard [38] and is done through one of these popular techniques - maximum likelihood estimate, maximum a posteriori estimate, expectation maximization, contrastive divergence, variational learning and Bayesian update. Majority of structure learning algorithms are unsupervised while parameter learning algorithms tend to be supervised or semi-supervised.

Inference in PGMs can be performed in an exact fashion with analytical techniques, or, approximately, through sampling-based or variational techniques. For exact inference, algorithms like Pearl's Belief Propagation in BNs and sum-product message passing algorithm in MRFs and RBMs are used [39]. For approximate inference, algorithms 'sample' – or extract points from – the probability distribution of the PGMs to perform inference. These techniques tend to converge to exact results with increasing number of samples. Few widely used algorithms are Gibbs' Sampling [40], Metropolis-Hastings algorithm [41] and several others, which together belong to the family of Markov-Chain Monte Carlo (MCMC) methods.

2.2 Emerging Nanotechnology

Another important aspect of ENAI are the various nanodevices which, through their physical characteristics such as electrical, magnetic, and optical behavior, provide strong foundations to the design and development of ENAI and associated circuit directions. We provide a brief technology primer regarding these nanodevices, which differ greatly in their properties and materials used, and are broadly categorized into the following device types: memristive, magnetoelectric, nanophotonic and emerging three-terminal devices.

2.2.1 Memristive Devices

Memristive devices are two-terminal passive nanoscale devices with pinched hysteresis voltage/current characteristics. The internal state (resistance/conductance) is determined by the history of applied voltage and current. They have a simple metal/insulator/metal stack (see Figure 1a). Because of their unique physical properties, fast and low energy switching, scalability, conductance modulation, they are one of the most promising technologies for ENAI. Memristive devices can be categorized based on their operating mechanism, physical properties, type of materials used etc. Based on filament rupture mechanism, there are two types of memristive devices namely drift memristors [42] and diffusive memristors [43] (see Figures 1a and 2a; based on type of switching material, the two main types are OxRAM and CBRAM; based on switching dynamics, they can be classified into linear and non-linear memristors devices. Figure 1a and 1b shows the memristive device stack and typical characteristic graphs. There are mainly two modes of operation, read mode and write mode. During the read mode, the conductances are sensed without disturbing their state, while during the write mode, the conductance is programmed by applying a voltage greater than the threshold of the device. The read and write voltages are encoded as pulse trains.

Phase-change memory (PCM) devices [47] are one of the more 'mature' emerging devices to date. They are often put under the 'memristive devices' category because they possess similar properties as

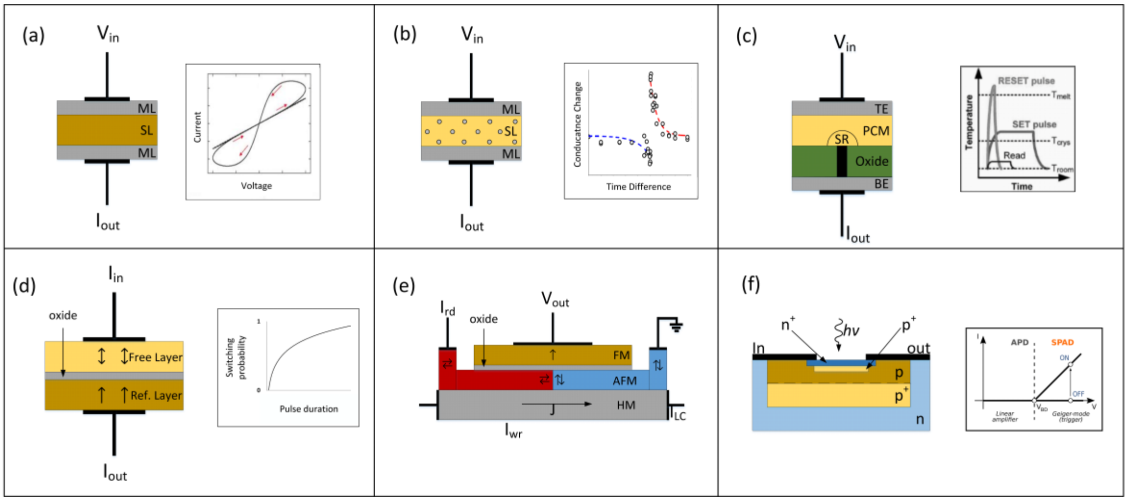

Fig. 2. Emerging nanodevices and their characteristics: (a) Drift Memristor; (b) Diffusive Memristor; (c) Phase-change Memory; (d) Magnetic Tunneling Junction; (e) Magnetic Domain-wall device; and (f) Single-photon Avalanche Detector

memristors. A thin layer of phase change material and insulator are sandwiched between two metal electrodes. They are simple two-terminal passive devices that utilize the phase transitions of materials to effect resistivity changes in devices. The phase transitions are from crystalline (low resistivity) to amorphous (high resistivity) phases. The conductance changes are affected through voltage/current pulses. Figure 1c shows a typical PCM device stack and its characteristic graph.

2.2.2 Magnetoelectric Devices

Magnetoelectric devices (see Figures 1d and 1e) are characterized by a multi-layer stack of magnetic materials and insulators. They operate on the principle of the (mis)alignment in the polarization of electronic spin in different metal layers separated by the insulating layer; if they are aligned, the device presents lower resistance, else the resistance is a function of the degree of (mis)alignment. The devices are programmed with one of two major methods: devices with three terminals are controlled by voltage applied to third terminal, which directly changes the polarization state. Devices with two terminals are programmed by use of spin-orbit currents, which apply spin-orbit torque on the material to change their polarization.

Magnetic Tunneling Junctions (MTJs): This family of devices typically consists of two magnetic layers separated with an insulator (see figure1d). One layer is permanently magnetized in a fixed axis. The other layer's magnetization is adjusted with various techniques to achieve different resistance values. MTJ devices can be both volatile and non-volatile. These devices have key properties such as very low power consumption, sub-nanosecond switching times, and non-volatility which make them ideal for use in several NEAI approaches. Some types of MTJs, based on their operating principles, are Straintronic MTJ [48], Spin-Transfer-Torque(STT) MTJ[50], Perpendicular Magnetic Anisotropy(PMA) MTJ [51], etc. Spin Torque Oscillators (STOs): This family of devices utilize the phenomenon of spin torque to generate controlled oscillations. The phenomenon of spin torque leads to oscillatory variation in the resistance of the device. These unique properties make them good candidates for applications that require nanoscale oscillators, such as oscillatory neural network architectures discussed in section 5.1.1. Some examples of STOs, typically named after their operating principle, are STT-STO [52], Giant Magnetoresistance(GMR) STO [53], Tunneling Magnetoresistance(TMR) STO [54] etc.

Domain Wall Devices (DWs): These devices utilize the existence of more than one domain of magnetization within the same ferromagnetic bulk. The boundary at which the various domains intersect is known as a domain wall (see figure 1e). This interface is usually mobile and can be moved by application of spin torque currents (J). Changing the position of domain wall changes the impedance provided by the device, and the domain wall position is static absent external sources. Applications of these devices include contiguous non-volatile memory, as synaptic elements in neuromorphic architectures, etc. Some examples of DW devices are Anti-Ferromagnetic DWs [55], and the skyrmion-motion based racetrack memories [56].

2.2.3 Nanophotonic Devices

These are devices that perform non-linear operations on light. Few examples which have been used in ENAI systems are Quantum-Dot LEDs (QD-LEDs) [57] and Single Photon Avalanche Detectors (SPAD, see figure1 f) [58][59]. The linear operations in optical domain can be done by use of passive elements like lenses. The optical devices allow for manipulations in all the properties of light – wavelength, frequency, phase and amplitude, which provides them with a rich representational framework. The devices are programmed mainly by application of a bias voltage. This bias voltage modifies the sensitivity of the non-linearity of the devices. These unique properties are useful in optical AI architectures (see section 5.2).

2.2.4 Multi-terminal Devices

In this category, some of the nanodevices with more than two-terminals that were not listed earlier are included. While two-terminal devices have the advantages such as simplicity in terms of connectivity, small footprint, multi-terminal devices provide more control over the conduction modulation mechanism and offer more degrees of freedom enabling more complex behavior. Some of the devices are MoS_2 FETs [60], Carbon Nanotube transistors (CNTs) [61], Nanoparticle organic memory field-effect transistors (NOMFETs) [62], Organic electrochemical transistors (OECTs) [63], Ferroelectric FETs (FeFETs) [64][65], Stochastic NMOS[66].

Table 1. Summary of demonstrated nanodevice sizes

Nanodevices	Smallest feature size demonstrated
Memristors	6 nm half-pitch and 2 nm critical dimension [44]
Phase Change Memory	Sub-10nm switching material thickness [45]
MTJ	14nm device with 75nm free layer radius [49]
Nanophotonic Devices	250nm spatial resolution of photon detection [59]
Multi-terminal Devices	5 nm thick gate dielectric [46]

3 Computational circuits for ENAI

These efforts are aimed at demonstrating fundamental aspects when designing architectures for AI using emerging nanotechnology. Hence, the focus is to demonstrate key computations of AI models using nanodevices rather than architecting a full-fledged system for AI from scratch. The works reviewed here address issues in one or more levels in the design hierarchy.

Vector matrix multiplication (VMM) aka vector dot-product is one of the most dominant computations in many AI models. Nanodevices arranged in a crossbar architecture can efficiently compute VMM operations by harnessing electrical properties governed by Ohm's and Kirchhoff's laws. This provides a scalable and compact way to realize ENAI. Typically, the VMM operation takes place between the input features (encoded as voltages/currents) and the parameters (encoded by physical property such as conductance, spin torque etc.) to produce outputs (voltages/currents) that subsequent modules can use for further processing. Some of the

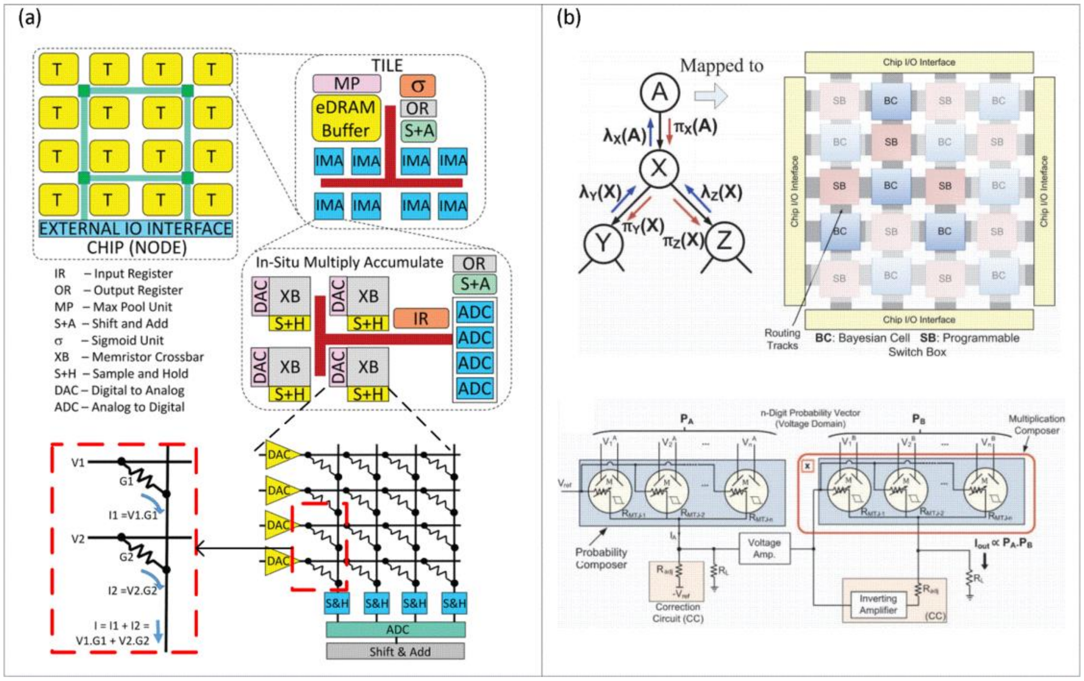

Fig. 3. (a) Example of a CNN architecture using memristor crossbars; Neurons mapped onto tiles which are connected with an on-chip c-mesh; vector-matrix multiplication using memristor crossbar [107]; and (b) Example of a BN architecture based on MTJs[115]; Structure of BN directly mapped onto reconfigurable fabric, probabilistic inference enabled by MTJ composer circuits.

physical properties of nanodevices can be varied in a continuous manner due to which the parameters can be modified overtime to *learn* the models. The parameters can be learned either ex-situ or in-situ. In ex-situ learning, the parameters are learned all at once using computer software and later mapped onto the crossbar. In in-situ learning, the parameters are still learned on a computer software but updating of the parameters takes place in stages after each epoch of learning. Apart from this, some works focus on training algorithms for crossbar-based circuits to account for non-ideal device characteristics [67], switching dynamics [68][69], eliminate noise during updating [70] variability [71]. This section reviews some of the works which have focused on demonstrating crossbar circuits to accelerate certain key operations in AI models. The focus is on demonstrating key aspects of such computing as opposed to realizing a full-fledged architecture for AI models.

Crossbars are used to design dot-product accelerators for neuromorphic and signal processing applications [72][73][74][75][76][77][78]. When synaptic weights of a neural network are mapped onto large crossbar arrays, all computations within a layer can be performed in a single step parallelly, thus achieving significant acceleration. Several works have focused on experimental demonstration of using crossbars for neural networks. Since the focus of these works is on demonstrating computations using nanodevice arrays, important peripheral functionalities like activation function, training etc. are implemented using external electronics or external computer running custom software. Many demonstrations involve showing fully connected multilayer perceptron network to do basic pattern classification [79][80][81]. Recently, recurrent neural networks such as LSTMs [82], Hopfield Networks [83] have also been demonstrated. While most of works mentioned above use ex-situ learning, few have demonstrated in-situ learning [84].

There has been work in use of crossbar architectures in enabling certain sub-class of probabilistic models with several architectural similarities to MLP NN models. In such models, namely Restricted Boltzmann Machines (RBM) and Deep Belief Networks (DGNs), while the computations in learning and inference operations are based on probability arithmetic, the architecture consists of a fixed layer-by-layer structure

Table 2. Summary of key demonstrations of core functionalities for enabling AI

Papers	**AI models**	**Device type**	**Key demonstrations**	**Key Results**
Suri et al. [86]	RBM	Memristor	Crossbar Circuit of OxRAM devices	device endurance: ~140 million cycles
Augustine et al.[95]	Generic	STT MTJ	variability-aware device simulations	Device choice depending on application type
Kim et al.[67]	DNN	Memristor	DNN framework for non-ideal IV	No dependency of accuracy on nonlinearity
Kataeva et al.[68]	FFN	Memristor	Modified backpropagation algorithm	State-of--the-art miss rate
Miao Hu et al.[72]	FFN	Memristor	Exp. demonstration of crossbar array of size 128 x 64	~90% accuracy with 6 bits precision
Preziozo et al[79]	FFN	Memristor	Exp. demo of ANN on a small crossbar network and in-situ training	100% classification accuracy for binary images
Burr et al.[80]	FFN	PCM	Exp. demo of large-scale ANN with PCM as synaptic weights	High classification accuracy of 82.2%
Li et al.[82]	LSTM	Memristor	Exp. demo of core part of LSTMs using crossbar circuits	Demonstration of classification and regression
L. Gao, et al[83]	CNN	Memristor	Demonstration of convolution kernel operation on resistive cross-point array	Experimental demonstration of kernel operations for edge detection

and operations resemble the multiply-accumulate operations of NNs. For this reason, most nanoscale implementations for RBMs and DBNs follow the crossbar architecture [86], the cross points encoding conditional probabilities instead of weights. Most approaches use memristor crossbar circuits, while a considerable few use MTJs. As with the NN approaches, these architectures share design similarities with CMOS-only architectural counterparts while mainly differing in their use of memristive or MTJ crossbars for multiply accumulate computations. The value proposition for these works lies in the fact that they provide experimental demonstration of various aspects related to crossbar-based computing.

4 Nanodevice-aware Architectures for AI

Re-evaluating the utility of nanodevices from being an efficient way to off-load some compute operation to being critical foundations to complete AI models leads to the emergence of these architectures. These are designed with tight integration of nanoscale devices and conventional CMOS and typically provide complete architectural support of AI models. The focus of these architectures is to best utilize the nanodevice properties for efficient implementations while designing architectures that are as close to the mathematical framework of their AI models as possible. The architectures are characterized with a hierarchical approach, with device-circuit-architecture co-design. Most of these works use simulations to showcase their work as often most of the technology aspects are proven by works reviewed in the previous sections.

Subsequent subsections will discuss these architectures in order of increasing functionality that they support. While some architectures support only inference or only learning over AI models, others support both.

4.1 Inference Engines

In this section, we focus on design of high-performance inference engines. Here, architectures implement inference algorithms using arithmetic circuits using nanodevices. Typically, nanodevice arrays are used for both storing as well as computing (compute-in-memory) on parameters of the AI models. Most often, these arrays are integrated on top of CMOS which provides support for other important functionalities such as activation function, sampling circuitry, signal restoration, timing and control circuitry, parameter update circuitry etc. Since the focus is on inference in these architectures, parameters are optimized externally (ex-situ) and then imported using parameter update circuits. Consequently, these systems are mainly geared towards providing support for static 'pre-trained' models. Hence, any training algorithm can be used to learn the synaptic weights which makes these architectures more versatile. This subsection is organized in two parts – architectures that enable inference in NNMs and ones which enable inference in PGMs.

4.1.1 Neural-Network Architectures

Several works have implemented mixed-signal inference engines for multi-layer feedforward neural networks and recurrent networks using memristive devices and phase change devices crossbar arrays. Weight matrices are mapped to the crossbar in case of MLP networks [99][100][101] while kernel filters are mapped in case of CNNs [102][103]. In case of CNNs, many techniques have been proposed to map the kernels to the crossbars. Additionally, spin-device-based convolution accelerators are also proposed [104] CMOS technology is used to implement the neuron and programming circuitries. Additionally, feedforward networks have been used to implement auto-associative memory [105]. [107] proposed a full-fledged mixed-signal memristor-based accelerator for deep learning. Memristor crossbars are used for storage and analog processing. It implements a tile-based pipeline architecture with each tile containing nanodevice-based multiply-accumulate units, CMOS-based activation function units and ADC units etc. Tiles are connected in a mesh network to provide full connectivity (see Figure 3). [108] proposes a novel architecture to implement machine learning algorithms. Memristors are used to implement CAM units to implement associative storage and processing. [109] proposes a memristor-based processing-in-memory (PIM) architecture to accelerate NN applications. It also proposes a software/hardware interface so that software developers can compile NN code to run on their accelerator. Recently, nanodevice-based computing-in-memory (CIM) chips have been proposed to realize large-scale NNMs [110][111]. These works aim to reduce the latency of multi-bit MAC

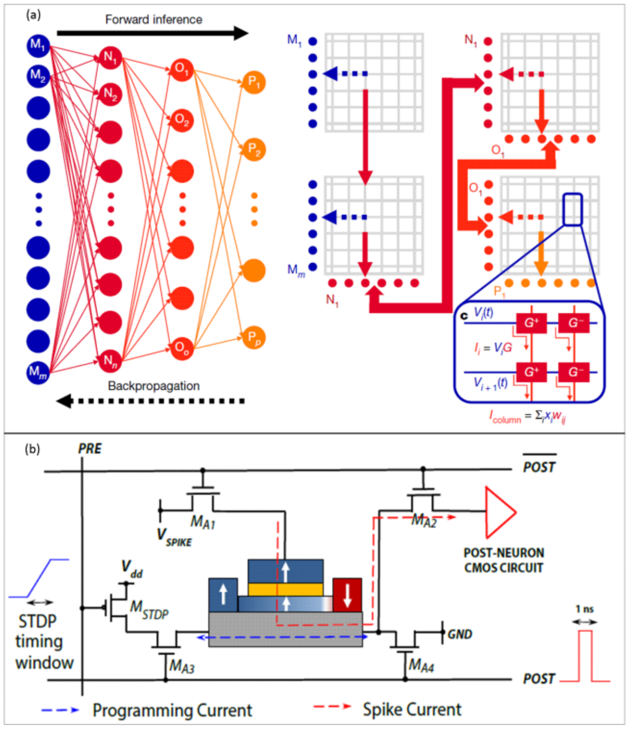

Fig. 4. (a) Fully connected neural network mapped into PCM arrays. Backpropagation algorithm which is a supervised learning algorithm is used to tune the weights[128]. (b) Spintronic Synapse with access transistors to decouple the programming and spike current paths to implement unsupervised STDP learning[131].

operations, improve accuracy of CNNs. To estimate and optimize performance of these memristor-based accelerators, several behavioral simulators have also been proposed [112][113].

4.1.2 Probabilistic Graphical Model Architectures

Several of these Architectures are focused on implementing PGMs, and typically consist of computational cells implementing a node of the graphical model and reconfigurable switchboxes to enable arbitrary connectivity. The computational cells contain the circuits required to perform the learning and inference operations for each node. These architectures follow as dataflow approach with asynchronous compute in each cell initialized by an update in their inputs. The node parameters are located within the cells in non-volatile memories, and circuits operate on these parameters and the inputs to update the node state. We shall now discuss these major architectural principles, as well as the benefits and challenges of the architectures that are representative of this approach.

At the higher level the architecture frameworks in this approach [114][115][116][117] tend to follow the design methodology of a uniform reconfigurable fabric. They consist of computational units(e.g., Bayesian Memory [114], Bayesian Cell [115][116][117]) augmented by programmable connectivity circuits. These architectures are primarily inference engines that implement the Pearl's belief propagation algorithm, while the structure and parameters of the BN model are learnt ex-situ (see, for example, Figure 3b). The belief propagation algorithm works by independent operations in each cell and passage of probability messages between adjacent cells. This is a departure from traditional approaches where the computations pertaining to each cell would be 'scheduled' to be performed sequentially, one at a time in CPUs, many at a time in FPGAs [119][120][121].

The computational cells comprise of two main parts- the cell parameters (known as Conditional Probability Tables, or CPTs) and the computation circuits. The CPTs and computations circuits utilize the low-power, non-volatile devices (e.g. memristors [114], MTJs [115]). The non-volatility of these devices allows for ultra-low-power storage; the co-location of computational circuits and storage mitigates any memory access latency. The computational circuits are designed to directly compute the mathematical operations involved in the belief propagation algorithm. The computations could be exact [114] [115][116] or approximate[117]. In some architectures, the devices perform the role of both memory storage as well as the computation circuit. Some architectures involve design of optimized probability encoding schemes using the nanodevices that benefit from the low precision requirements of PGM applications [115][116], while others have designed probability encoding circuits that provide scalable precision [117].

These architectures are evaluated from circuit level all the way up to application level. The various innovative designs, optimizations at circuit and architectural level, as well as the low-power nonvolatile devices together result in significant gains over conventional approaches. Power-performance benefits of up to 5 orders of magnitude are reported compared to traditional approaches like software implementation on 100-core processors [115] and two orders of magnitude over ASIC implementations using conventional devices[119].

4.2 Architectures with Support for Learning

Training is the most computationally expensive aspect in any AI model. AI models implemented using conventional technology with multiple GPUs require power in the order of several 100s to 1000s watts of power for training [122]. Since nanodevices enable in-memory computing meaning computations often take place at the location of the synaptic weight storage. Hence training AI models implemented using nanodevices can be significantly faster. Nanodevices are scalable, provide fast and low power switching dynamics which is a crucial aspect for implementing learning systems. The techniques how these parameters are tuned depend on the type of AI model, type of learning technique, etc. The central idea is to tune the physical properties of the nanodevices (conductances, magnetoresistance etc.) which implement the free parameters of the AI models. These characteristics are ideal for supporting learning in technologies that implement AI models. Owing to these benefits of employing nanodevices, several works have focused on designing architectures/circuits to support learning. This next subsection reviews several works which are aimed at providing learning support for AI models.

4.2.1 Supervised Learning

The overarching motivation for these works is to design compact learning cells while still maintaining high accuracy for training. AI models are often trained using a popular supervised training algorithm known as *backpropagation*. It involves propagating errors backward through the network layers to update the weights based on gradient descent. Hence, it involves a lot of complicated computations and caching of intermediate data. There are several challenges when designing custom hardware learning frameworks using emerging technology for AI. Since implementing a fully-fledged backpropagation algorithm in hardware is expensive, it is modified so that custom hardware using CMOS technology can be employed. Several works have proposed acceleration frameworks for providing supervised learning support to multilayer feedforward neural networks [123][124][125][126][127][128][129]. Better yet, some works have proposed learning cells using emerging technology which are compact and consume less power than CMOS hardware. If the framework is mixed signal, caching of analog signals is a problem. [130] implements clever caching techniques to store intermediate results. Tuning process for nanodevices is complicated, time consuming and may require a lot of hardware resources.

4.2.2 Unsupervised Learning

Several AI models such as Spiking neural networks, Gaussian Mixture Models, etc. use unsupervised learning algorithms such as Hebbian learning, spike timing dependent plasticity (STDP), Expectation-Maximization, Bayesian Inference, Contrastive Divergence etc. for updating the weights. These algorithms sometimes draw their inspiration from specific aspects of how the brain implements learning and how this learning relates to Information-theoretic concepts. Networks which implement unsupervised learning that make use of these algorithms are dynamical in nature, i.e., the model performs learning *while* performing inference, in contrast to supervised learning, where learning takes place beforehand. These learning algorithms are predominantly used in spiking neural networks and RBMs although other types of networks have also been shown to use them. The continuous learning allows these models to adapt to changes in real-time making them flexible and robust. Since unsupervised algorithms are dynamical in nature, nanodevices which have low programming energy and fast switching characteristics are most suited.

Works in this domain are digital or mixed-signal in nature, and primarily focus on demonstrating unsupervised learning using nanodevice-based circuits. [131] proposes a hybrid spintronic-CMOS SNN with on-chip unsupervised learning support. Low programming energy and fast programming of spintronic devices make them ideal to implement STDP learning. The digital architecture allows for reconfigurability which makes it flexible enough to implement a host of models. The weights are stored in analog fashion and converted to digital using ADCs. A digital pulse-width-modulation scheme is used to tune the memristive devices to implement learning using STDP [132]. A semi-supervised learning circuit framework for domain-wall MTJ based neural network architectures is proposed [133]. A comprehensive memristor architecture

Table 3. Summary of key Nanodevice-aware Architectures for AI

Papers	AI models	Device type	Key contribution	Results
Yakopcic et al. [102]	CNN	Memristor	Completely parallelized inference architecture for CNN	High classification accuracy on MNIST dataset
Yong Shim et al.[104]	CNN	Domain-wall	Hybrid spintronic-CMOS design for convolution computing	~ 2.5 x lower energy vs CMOS-only implementation
Xiaoxiao Liu et al.[105]	ANN	Memristor	Reconfigurable architecture neuromorphic accelerator	~ 2 orders performance, 2 orders energy vs. CPU
Ali Shafiee et al.[107]	CNN	Memristor	A full-fledged pipeline architecture for CNNs	~ 1 order of magnitude over state-of-art previous ASIC implementation
Ping Chi et al.[109]	ANN	Memristor	NN accelerator with hardware/software interface	~ 3 orders of magnitude performance, ~2 orders energy
Raqibul Hasan et al.[123]	FFN	Memristor	On-chip backpropagation training of crossbar arrays	Energy efficient and compact neuro systems
Djaafar Chabi et al.[124]	FFN	Memristor	compact learning cell design for high density integration	acceleration of learning and high area density
Daniel Soundry et al.[126]	ANN	Memristor	Compact learning cell design for low-power learning	2% and 8% of the area and static power compared to CMOS-only approaches
Abhronil Sengupta et al.[131]	SNN	STT-MTJ	architecture and circuits for learning using STT-MTJs	SNN for MNIST digit recognition with ~ 48fJ programming energy
Zaveri et al.[114]	BN	Memristor	Exploring CMOS/Nanoscale Integration in architectures	Comparison of digital and mixed-signal architectures for BNs
Khasanvis et al.[115]	BN	S-MTJ	Circuit/Architecture design and large-scale evaluation	~6000x power-performance benefit vs. 100-core CPU
Kulkarni et al.[117]	BN	S-MTJ	Scalable-precision approximate Compute architecture	~30x area reduction for high-precision applications
Nasrin et al.[136]	RBM	STT-MTJ	Stochastic Computing for online learning	sub-picojoule energy per neuron operation
Behin-Aien et al.[139]	RBM/ BN/Ising	MTJ	belief unit' for building graphical model architectures	simulated a BN application
Bojnordi et al.[134]	RBM	Memristor	Online leaning circuit/architecture design	~100x performance vs. CMOS

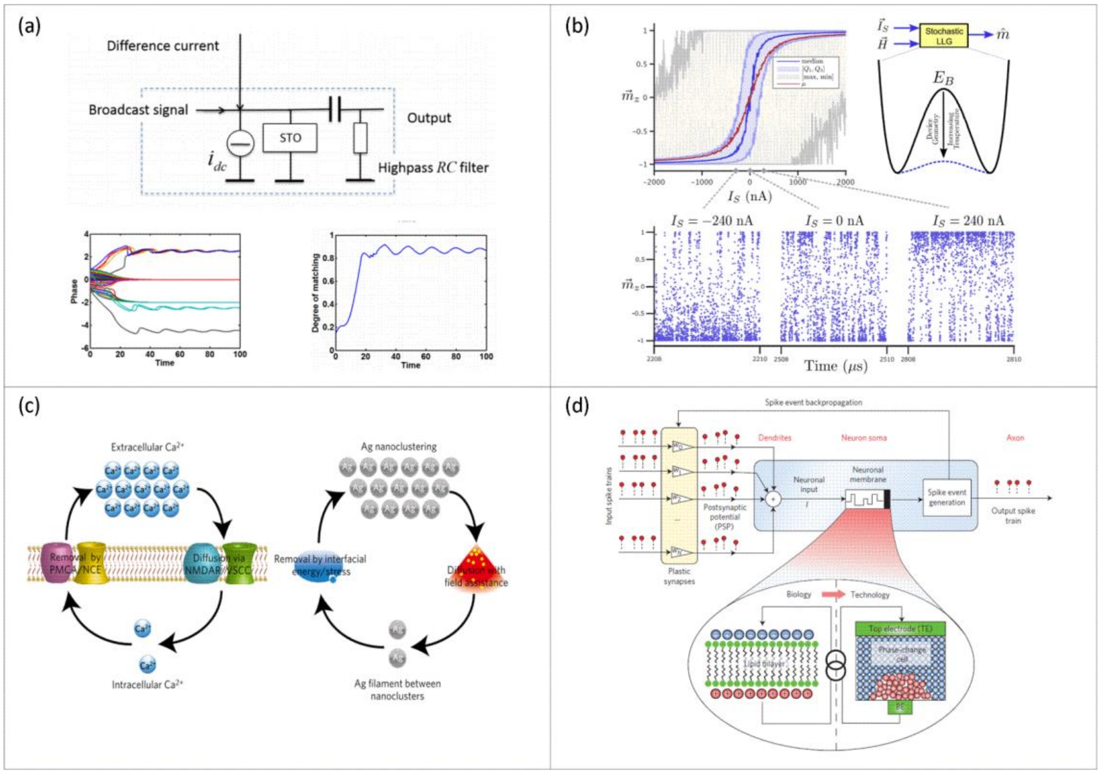

Fig. 5. Key functionalities with nanodevice circuits. (a) Coupled oscillation behavior in Spin-torque Oscillator device-based circuit[143]; (b) Stochastic behavior of MTJs when exposed to spin-orbit current[151]; (c) (left) Diffusion of Ca^{2+} in neuron cell membrane (right) Ag diffusion dynamics in diffusive memristor which demonstrates short-term synaptic plasticity [161]; and (d) Artificial neuron based on a phase-change device [173].

that performs optimization and learning in RBMs through Bayesian implementation of contrastive divergence is proposed [134]. [135] proposes a similar learning framework, but with PCM devices and related support circuits. [136] introduces a learning and inference computational unit based on MTJs that the work proposes could be used in several PGM models such as BNs, RBMs and Ising models. The work supports several learning methods by mapping them into the contrastive divergence framework.

Several works shown in this subsection, and more specifically [133][134][137][138][139] indicate, the various unsupervised learning methods implemented in hardware are algorithmically equivalent to each other. Given that some algorithms drive inspiration from the workings of the Brain while others are based on probability arithmetic and Bayesian statistics, their equivalence is of great importance in the design of architectures for AI models. This equivalence could lead to different model types being learnt using these generalized learning techniques, and further toward implementation of hybrid AI models such as Variational Autoencoders [140] and Bayesian Neural Networks [141] in future architectural design efforts.

5 Toward All-nanodevice architectures for AI

With the advancement of device development, several recent works focus on both the design of key functionalities and entire computational frameworks of AI models using only emerging and unconventional devices, with minimal, or in some cases, no CMOS circuitry partaking in the core functionality. This section is organized in two subsections – architectures focusing on key functionalities and architectures implementing all-nanodevice frameworks for AI.

5.1 Architectures Enabling Key Functionalities with Nanodevice Circuits

AI models typically consist of several mathematical operations. These operations could be of various forms and originate from various mathematical disciplines such as calculus, algebra, probability theory etc., and are typically computationally intensive. The distributed nature of AI models entails that these operations need to be performed at each node of the graph, further increasing the computational complexity for large applications. The complexity of these operations is evident in the large number of clock-cycles required in case of software implementations and the number of devices and circuit configurations in conventional hardware implementations. Research in material science and device physics has led to discoveries of a variety of nanodevices with a departure from the traditional 'switching' behavior of transistors. Through the process of design and fine-tuning of these devices, they are made capable of performing the complex mathematical operations intrinsically with little to no external circuits required. This section shall discuss the research directions which demonstrate the mathematical operations used in AI models enabled by novel nanodevice behavior.

5.1.1 Coupled Oscillations

These circuit designs attempt to model the coupled-oscillatory behavior of neuron spikes. The mathematical framework of coupled oscillatory systems and their use in machine learning applications is detailed in [142]. The circuit designs in this approach use nanodevices (mainly various devices in STO family) to obtain the mathematical framework of coupled oscillatory behavior in-circuit. The models obtained from these circuit styles are called oscillatory networks, or oscillatory neural networks. These circuit designs are shown to demonstrate operations like pattern recognition [143] and classification [144][145]. The coupled oscillator phenomenon allows for small number of coupled oscillators to perform comparably with larger traditional NNs (see Figa).

The nano-oscillators along with their support circuits form a phase-locked loop (PLL). These PLLs are then connected to each other and initialized with random or preset frequencies or phases. The input is applied to the configuration of PLLs as frequency or phase perturbations through frequency-shift or phase-shift keying respectively. The PLL configurations get coupled and resonate with a certain frequency. The label in which in the inputs are classified is encoded in the resonant frequency.

5.1.2 Distribution Sampling and Stochastic Behavior

Approximate inference is widely used in PGMs for its simplicity and any-time nature. The core operation required in approximate inference is sampling from a distribution. When a PGM is evidenced upon a certain subset of its random variables, it forms a conditional probability table or distribution (CPT, CPD). Sampling is the process of obtaining sample values of the remaining random variables that correspond to this conditional distribution. There have been approaches to use memristive, magneto-electric [134][149][150][151][152][153][154] and nanophotonic [155][156][157][158] devices and circuits to encode and sample from distributions. The distributions that have been shown to be sampled from include both discrete (categorical) [150] and continuous (Bernoulli, exponential, Gaussian, uniform etc.,) [155][157][156].

The circuit design in these approaches usually involves the design of 'sampling units' (e.g., Resonance Sampling Unit [155], Stochastic Bayesian Node [150]). These sampling units produce 'samples' in the following way: in the discrete case, each state of the random variable is sampled proportional to its probability as encoded in the CPT; in continuous case, a sample is produced from the distribution parameterized by the current CPD, such that a sufficiently large number of samples will resemble the CPD. These sampling units are then arranged in graphical structures and can perform independent sampling based on evidence obtained from neighboring nodes.

These sampling processes, due to their use of device physics for sampling and their massive parallelism due to the use of distributed sampling units, are potentially orders of magnitude faster and more power efficient than software and conventional hardware approaches (see Fig5.b).

5.1.3 Synaptic Dynamics

Synapses are the fundamental entities of learning and memory in the biological brain. Spike timing based synaptic plasticity is widely believed to be the basic phenomenon behind learning and memory. This plasticity is achieved by the modulation of complex electrochemical activity in the synaptic clefts. Building circuits using conventional technology to emulate this complex electrochemical behavior is expensive.

Table 4. Summary of key works toward All-Nanodevice Architectures for AI

Paper	**AI models**	**Device type**	**Special property**	**Results**
Nikonov et al.[143]	Associative Network	STO	Coupled Nano-Oscillations	Unsupervised pattern recognition with 64 oscillators
Sengupta et al.[146]	ONN	STO	Spin-based Neuronal Activation	Sub-picojoule neuron activity vs. ~700pJ CMOS neuron
Kulkarni et al.[150]	BN	STT-MTJ	Probabilistic switching of magnetization	~86,000x speedup vs. software baseline
Sutton et al.[151]	Ising Model	STT-MTJ	Stochastic nano-magnetic behavior	Demonstrate several NP-Hard problems using nanodevices
Onizawa et al.[152]	BN	MTJ + Transistor	Probabilistic behavior of composite device	BN 'translated' to probabilistic logic for inference
Zand et al.[154]	DBN	Sigmoidal STT-MTJ	Sigmoidal behavior for neuronal activation	4-layer DBN trained on MNIST with 97% accuracy
Wang et al.[155]	MRF	QD-LED + SPAD	Resonance Energy Transfer Behavior for Gibbs sampling	Image segmentation and motion estimation, ~40x speedup vs. GPU
Blanche et al.[158]	BN/MN	QD-LED + SPAD	Optical behavior for emulating functions (log, exp, etc.)	Demonstrated VMM, exp and log functions
Mirzhai et al.[159]	Basis Functions	STO	Population encoding in oscillatory networks	Demonstrate brain-like auditory behavior
Tuma et al.[173]	SNN	PCM	Artificial stochastic neuron which models membrane potential	Temporal integration in nanosecond timescale
Torrejon et al.[144]	ONN	MTJ	Nonlinear oscillators	Ex demo oscillators to achieve spoken-digit recognition accuracy similar to that of neural networks
Schnerider et al.[164]	SNN	JJ	fast, low-power stochastic synapse	Simulated a basic neuromorphic circuit with a neuron and synapse
Wang et al.[161]	SNN	memristor	Diffusive dynamics closely resembling influx and extrusion of calcium ions	Ex demo of STP and LTP
Pickett et al. [172]	SNN	memristor	spiking behavior similar to HH axon	Ex demo of spike trains and thresholding
Sharad et al.[175]	SNN	STT-MTJ	low-current, low-voltage, high speed switching for thresholding	2 orders of magnitude low energy
Sengupta et al.[166]	STDP	STT-MTJ	STDP like behavior	pico-joule level energy consumption per synaptic event

Recently, emerging nanodevices with properties similar to biological synapses have been investigated. As nanodevices have low footprint and operate with very low power, synaptic implementations using these nanodevices achieve substantial reduction in complexity when compared to conventional technology. This section reviews works which use nanodevices to emulate fundamental aspects of synaptic behavior thus paving way for all-nanodevice architectures for AI.

In [160], a nanoscale memristive synapse was demonstrated for the first time ever showing synaptic functions such as Spike Timing Dependent Plasticity (STDP). [161] developed a class of memristive devices called as diffusive memristors whose diffusive Ag-in-oxide dynamics closely resembled calcium dynamics in biological synapses. Biological learning and memory mechanisms such as STP, LTP, STDP were emulated with these devices paving way for more robust hardware implementations of neuromorphic functionalities (see Figure 4c). [162][163] reported a phase-change material-based synaptic element that mimics biological synapses such as synaptic learning rule using continuous resistance transitions in the material. [164] showed a synaptic emulator based on a dynamically reconfigurable low-energy JJ device capable of non-Hebbian learning. [165] demonstrated an inorganic synapse which emulates important synaptic functions such as STP and LTP. [166] demonstrated a spin-orbit torque-based device which implements STDP with pico-joule level energy consumption per synaptic event. Very recently, multi-terminal devices have been developed to emulate synaptic dynamics. [167] demonstrated an analog synapse learning to accelerate DNN training using three terminal devices.

5.1.4 Neuronal Dynamics

Artificial neurons are the fundamental building blocks of neural networks along with synapses. Neural functionality ranges from implementations which are inspired by biology to implementations which mimic biological counterparts. Neurons which are inspired by biology typically are simple in nature while the ones closer to biology typically exhibit complexity. The type of neuron functionality depends on the architecture of the neural network and application targeted. Some of the examples include Hodgkin-Huxley model [168], Integrate and Fire model [169], Hindmarsh-Rose model [170], McCulloch-Pitts model [171] etc. Since neurons exhibit complex functionality, implementing their behavior using software or CMOS systems proves to be very expensive in terms of performance, power, and area. In addition to this, seamless integration between synapses and neurons becomes a problem. Hence, scalable realization of neurons is one of the fundamental challenges in realizing hardware systems for AI. Several nanodevices with device properties have been proposed in the past decade which exhibit many of the complex behaviors, if not all of them. In this section, we review some of the works which propose new devices which have the potential to replace large amount of associated complex circuitry.

[172] proposes a Hodgkin-Huxley neuron using Mott memristors. [173] proposes a phase-change-device-based integrate and fire neuron with stochastic dynamics. The device realizes many attributes like membrane potential stored in the form of phase configuration, phase transitions on a nanosecond timescale, stochastic phase transition etc. (see Figure 4d). [174] proposes a integrate and fire neuron using memcapacitors. [175] demonstrates a spin-torque-based neuron which mimics the analog summing and thresholding behavior with high energy efficiency. [176] demonstrates a domain-wall MTJ based neuron with leaky integrate and fire behavior.

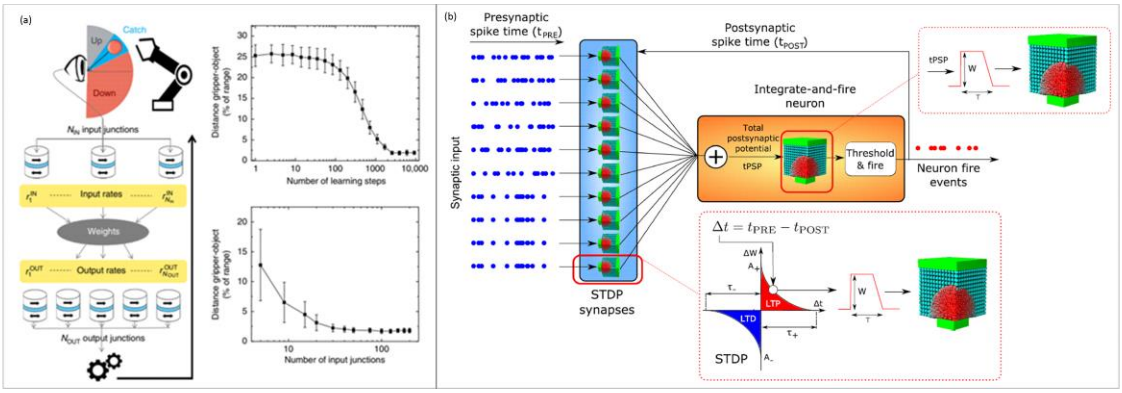

Fig. 6. All-nanodevice NEAI approaches. a) An all-MTJ system to automate learning process of robotic sensorimotor control using oscillatory circuits implementing non-linear basis functions[159]; b) Schematic illustration of an all-memristive computational primitive. The neuronal membrane potential and the synaptic weights are emulated in the phase configuration of nanoscale phase-change devices[177].

5.2 Synergistic Device-Circuit-Architecture Design

In this subsection, we discuss some approaches toward ENAI that involve an end-to-end design using nanodevices with minimal support from CMOS technology. These systems are purpose-built from ground up to enable/accelerate specific AI models, and involve design of circuits using nanodevices special-purpose circuits and customized large-scale architectural features that synergistically work toward inference and learning operations of an AI model. These approaches are fewer in number than the previous two approaches and could be regarded as a second-generation of ENAI systems. In this subsection, we mention few such approaches which are representative of the work in related directions. [177] demonstrates an all-memristive SNN neuromorphic architecture with phase-change memristors for implementing the functionality of synapses and neurons. [178] lays out a vision for an all-spin neural network architecture with various spintronic devices implementing neurons and synapses. In [159], an Oscillatory Population-Encoding Architecture is proposed for Sensorimotor Control: this work attempts to reconstruct the computational properties of a population of neurons encoding a non-linear basis function. They achieve this by designing a ‘population’ of STO devices which are then tuned to learn a non-linear basis function defined as a mapping between input (injected) frequency and output (resonance) frequency. Although the work still contains CMOS support circuitry, it is merely peripheral, the core functionality for learning is completely spin-based. [179] lays out a vision for an all-photonic spiking neural network using a phase-change resonator. [158] envisions an all-photonic PGM architecture with individual nodes capable of performing sampling-based inference tasks using optical nanodevices.

6 ENAI Technology fabrics

This section focuses on works which aim to create next-generation integrated circuit technology for AI by incorporating novel nanodevices and their associated material stacks. A vast majority of the nanodevices are passive devices and hence cannot be used for implementing general purpose logic. Because of this, even all nanodevice architectures would need to rely on digital/analog CMOS logic circuits. In addition to this, there are other important aspects of IC technology such as circuit, placement and routing, thermal management, variability, manufacturing pathway etc. ENAI fabrics are aimed at solving most these aspects. These fabrics are the final logical step for realizing hardware systems for AI. Some works focus on integrating nanodevices with CMOS technology while others focus providing a complete solution for all technology needs. These are enabled by new fabrication procedures that allow for hybrid integration, 3D vertical integration of components to achieve connectivity, and manufacturability with minor changes to the existing fabrication processes. These fabrics and technologies provide a major step towards enabling broader assimilation of ENAI into the mainstream in the future.

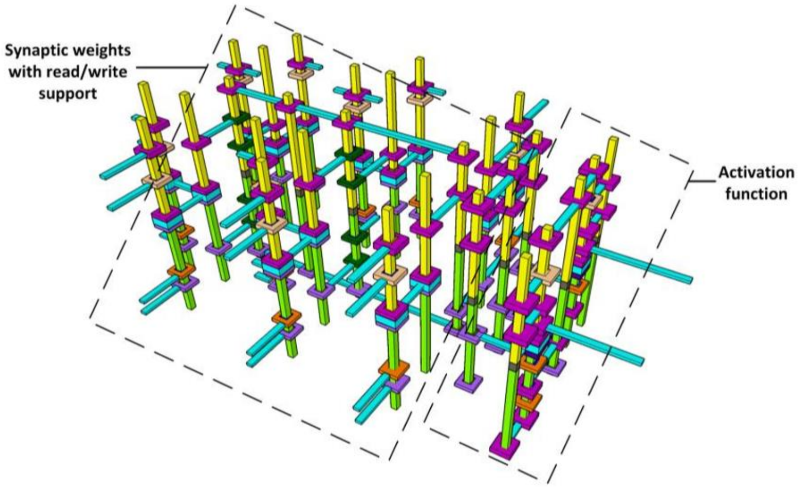

Fig. 4. 3D spatial distribution of synaptic weights and neuron in SkyNet fabric [182]. Fabrics purpose-built for AI applications like these can spawn the next generation of ENAI.

The most common implementation strategy is to do a nanodevice/CMOS hybrid integration. In these implementations, nanodevice crossbar arrays are complemented with conventional CMOS substrate through Back end of Line (BEOL) integration [131][180][181]. A 2D array of vias provides electrical connectivity between the CMOS and crossbar arrays. Since CMOS technology is very mature, these fabrics can rely on it for implementing key aspects of the design, peripheral circuitry and signal restoration. [182] proposed a new 3D integrated ASIC technology for NNMs. Instead of using an incremental approach of stacking, it uses fine-grained 3D connectivity between the nanodevices and transistors. Hence, it allows for 3D spatial distribution of synaptic weights and neurons and interconnect (see Figure 5). The ultimate vision of such fabrics is to cater to all aspects of IC technology for implementing AI.

7 Discussion and conclusion

We have reviewed the domain of artificial intelligence with emerging technology and divided the work according to their intellectual contribution. Emerging technology offers a lot of advantages over conventional technology for efficiently implementing architectures for AI. In this section, we discuss the potential of ENAI, key challenges and limitations that need to be overcome for it to become mainstream.

7.1 Potential of ENAI

Most works in ENAI discussed earlier include performance and power consumption analysis. These indicate anywhere between 2-4 orders of magnitude benefits vs. certain conventional reference designs. While these are useful as evidence for the potential for those specific works, they may not be as useful regarding demonstrating the capabilities of ENAI in general. Furthermore, it is important to extrapolate these individual results to the scale and constraints of future real-world workloads, which could provide a better point-of-view for the impact of ENAI on the energy and performance scales associated with computational requirements of AI applications. To demonstrate this potential of ENAI, we design few workload scenarios that are representative of current and near-future computational needs of AI applications in both large-scale as well as low-power learning and inference tasks. We estimate the power and performance numbers for software/GPU approach with the FPGA/ASIC and ENAI approaches. The ENAI approaches are further divided into two versions – CMOS with ENAI acceleration and all-nanodevice ENAI with minimal CMOS circuitry. The former represents the short-term future architectures while the latter represent the more distant future architectures. In coming up with the estimates, we make certain assumptions. We assume linear scalability for numbers suggested in literature for scaling up from reported applications to the applications we consider. Having made these assumptions, we report conservative numbers. To reach these numbers, we estimate the computational speed in operations-per-second and power as watts-per-operation as implied by the reported numbers in papers, and then scale them to the operations required by the applications listed below.

7.1.1 Large-scale image recognition Training and Inference

Computer vision is a domain that has seen great developments thanks to the advanced neural network architectures, especially convolutional networks. Development and widespread use of these models is dependent on the ability of rapid prototyping and learning from huge datasets. With growing use of this approach in mainstream, there is a substantial requirement of high-performance hardware solutions for newer and more demanding workloads. We consider a large-scale ImageNet [8] CNN with 60 million parameters, trained on 1.2 million examples from the ImageNet database. We have considered the current state-of-the-art single Tesla v100 [12] (130 TFLOPS) GPU for the baseline. For ASIC metrics, we have considered the DaDianoNao architecture [106] which uses near data processing approach using eDRAMs for memory and digital CMOS for compute. The authors have reported their speedup and power savings vs. Nvidia K20M (3 TFLOPS) GPU for the ImageNet model. Finally, we estimated the speedup and energy benefits of DaDianoNao as compared to Tesla v100. For ENAI, we consider ISAAC [107] which is a CNN accelerator which uses memristor crossbars for multiply accumulate operations and digital CMOS for neuron functionality. We chose this design because of two key reasons. First, this is one of the few full-fledged architecture for CNNs using nanodevice crossbars integrating several digital and analog components. Second, the authors have benchmarked several CNN architectures including an architecture with 330 million parameters. Even though the authors have not compared their design with GPUs, they have reported the speedup vs. DaDianoNao which forms the basis for our estimation vs. GPUs. Table 4. summarizes the power and performance estimates for implementations based on conventional technology such as GPUs and ASICs and implementations based on CMOS with memristors. In this table, we report relative speedup and energy benefits but not absolute numbers such as TFLOPS and TFLOPS/W because these raw numbers don’t translate to actual performance and power benefits. Instead actual benefits depend on the type of application being implemented, the number of basic operations involved, number of shared memory accesses (for GPUs) involved etc. We provide these numbers by directly referring to the above papers since they have already reported the numbers and are model specific.

Table 5. Estimated Energy and Speedup of various platforms for large-scale image recognition model

	Energy Benefits	**Speedup**
GPU (Tesla v100)	1x	1x
ASIC (DaDianoNao)	75x	10x
ENAI (ISAAC)	90x	140x

7.1.2 Discovering genetic networks at whole genome scale

PGMs like BNs have been widely used in life sciences, especially in the data-intensive domains of bioinformatics and computational genomics. Progress in genomic and proteomic sequencing tools has led to an abundance in data but implementing whole genome scale networks with BNs is prohibitive with current technology and can only be done after setting constraints on the design. We consider one such application which is part of the DREAM challenges, which are well-known benchmarks in bioinformatics [192]. This involves performing probabilistic inference on genome scale networks (3,456 genes, 300 experiments). One key difference in this application compared to the ImageNet model is that the computation involved is not standardized; in fact, several attempts for this challenge typically use several different algorithms ranging from brute force to very complex, making this analysis more challenging. Hence, we compare the hardware platforms with respect to the actual number of CPUs used by one of the leading approaches for this challenge, and then scale it to other platforms in roughly equivalent compute. For hardware platforms being compared, we omit GPUs here as to the best of our knowledge, there are no well-known GPU accelerated inference implementations for problems at Genome-scale and which use non-trivial algorithms for inference typically required for such applications. The original work reported numbers for CPU, and we extrapolate numbers for FPGA/ASICs based on Bayesian inference accelerator works in those hardware platforms [193][194]. The metrics being reported for this application aim to capture the large-scale power and performance differences that could be achieved with NEAI for very large-scale applications frequent in life sciences and other such disciplines. The results are summarized in Table 6:

Table 6. Estimated Power and Performance of various AI implementations for discovery of genome-scale BNs

	Est. Power (Watts)	**Est. Runtime (Hrs)**
CPU	30k-40k	200-300
FPGA/ASICs	12k-20k	50-80
ENAI (CMOS+MTJ)	~3k	~50
ENAI (All MTJ)	~1.5k	~25

From multiple works explored in this survey, it is evident that Magnetoelectric devices are more suited for PGM-type workloads. For estimating the performance for these ENAI approaches for the genome-scale BN workload, we choose i) CMOS-MTJ and ii) All-spin MTJ architectures respectively to capture both hybrid all all-nanodevice approaches to BNs. The results are extrapolated from the average power and performance numbers reported in corresponding sections and are rough estimates. These results suggest that, based on the work so far in ENAI, there is a significant improvement of 1-2 orders of magnitude in power and performance estimates for large scale AI applications. These applications are indicative of the near-future workloads of AI systems. With such large-scale applications being commonplace, the efficiency and performance improvements very well warrant the efforts in developing and innovating ENAI research efforts. The next section shall discuss the limitations of ENAI, and the challenges involved in further development of ENAI.

7.2 Impact of Device Variability and Yield

Some of the issues facing nanodevice arrays are variability and yield. Initially, device endurance was an issue, but several works have demonstrated devices with very good endurance since the beginning of this decade [86][87]. Advantage of using nanodevices for architecting for AI is that the imperfections in devices can sometimes be used to our advantage. AI models are in general more tolerant to device related issues. In this subsection, we review some of the works which try to solve some of the issues associated with these nanodevices. Some of the works have studied the impact of device variability on classification accuracy in neural network models. [88] discusses the impact of device variability on the performance of feedforward neural networks. The authors randomly drew samples of high and low resistance values of each device from a log normal distribution and still achieved recognition accuracy of over 97% for image recognition tasks. Several algorithms have been proposed to determine conductance values such that NNMs can tolerate variations in device dimensions [89][90][92]. These algorithms are collectively known as variation-aware training algorithms. Another way to overcome device variation and low-yield rate is by having multiple parallel nanodevices to store a single synaptic weight [91]. Since nanodevices sub-5nm dimensions have been shown, the overhead resulting from using multiple devices is small. Several works have also studied the impact of device yield rates on recognition accuracy in NNMs. For example, [93] reports that even with 90% device yield, CNN can still achieve over 96% recognition accuracy. Several works [95][96][97] discuss the design of MTJs to mitigate variation and to perform variability-aware device simulations [98] for their use as non-volatile memory and in AI applications. These works allow for developing NEAI architecture, where the well-studied noise and variability of these devices is used to perform stochastic computations. For example, the analysis in [98] shows that the variability in MTJ devices can be modeled accurately by a skew normal distribution, and using this distribution, the circuits can be designed to a target write-error-rate (WER), where if the application can allow a higher WER (by just 1% more), the write voltage can be reduced from 1.1v to 0.8v, reducing write energy by ~37%.

7.3 Challenges and Limitations

Although emerging technologies offer a lot of potential for architecting AI systems, they still have many challenges and limitations which impede their deployment. Individually evaluated, nanodevices have several advantages such as compactness, unique properties, low-power etc. but demonstrating these advantages through large-scale applications still needs some work. Researchers need to come up with innovative circuit and architectural ideas to utilize the special properties of nanodevices. Although issues have been addressed individually, whether they would be issues when systems are built is another question.

Wafer-scale manufacturability is one of major concerns for systems with nanodevices. Fabs must spend significant amount of resources to develop and implement new processes to manufacture these nanodevices with high yield rates. This can happen only if companies believe that using these nanodevices give significant benefits. One of the main goals of this paper is to show that these nanodevices indeed have huge potential to substantiate these costs. It is going to be the case in the coming decade as the innovations in CMOS architectures start to saturate toward diminishing returns and these ENAI approaches are perceived to be more viable.

Another major concern in the immediate future as the CMOS-augmented ENAI architectures start to become mainstream is the problem of signal conversion overhead. Typically, analog CMOS designs are used for very specific applications in signal processing applications. Digital CMOS technology is mainstream for designing processors, accelerators, FPGAs, ASICs etc. On the other hand, nanodevices mostly operate in analog domain. Hence when emerging devices are combined with CMOS, signal conversion between the two domains becomes an overhead. This issue may be overcome in the future when end-to-end all-nanodevice architectures on the lines of as described in section 5 can be designed.

Software integration is another area where there has been little work. As the ENAI architectures become more mainstream, the ability for developers to simply use the popular libraries such as TensorFlow, PyTorch, R in writing their models which then run on this emerging hardware would be a challenge. This would require design of sophisticated hardware-aware compilers which refactors and distributes the computation to get the most out of these high-performance platforms. This tells that there is a still a long way to go before computers fully built using nanodevices come to existence. In the short term, one possible way to go forward would be to design these nanodevice-based PCI-e cards that can slot into the motherboards and have software recognize them as co-processors or accelerators.

7.4 Conclusion

ENAI encompasses directions, works, and efforts that focus on designing AI architectures and associated circuitry leveraging unique properties of emerging nanodevices. ENAI has enormous potential to accelerate AI research which could trigger a wide-scale adoption in real-world applications. As unique device technologies become more prominent, AI algorithms must rely more on their capabilities. This, to not only better utilize the nanodevices but also to inform research in emerging technology and nanoarchitectures, of specific opportunities. As we have uncovered in this survey, emerging nanotechnology promises many orders of magnitude power and performance benefits vs conventional directions. It is likely that future directions will rely increasingly on nanodevices. On the other hand, conventional CMOS-based technology has still untapped benefits for AI; further engineering and research in AI-specific custom ASICs, AI-related instruction set extensions for microprocessors, and hybrid approaches as well, are to be expected in the coming decade.

REFERENCES

[1] Kyle Hollins Wray, Stefan J. Witwicki, and Shlomo Zilberstein. 2017. Online decision-making for scalable autonomous systems. In *IJCAI International Joint Conference on Artificial Intelligence*. DOI:https://doi.org/10.24963/ijcai.2017/664

[2] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. 2015. DeepDriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*. DOI:https://doi.org/10.1109/ICCV.2015.312

[3] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B. Choy, Philip H.S. Torr, and Manmohan Chandraker. 2017. DESIRE: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. DOI:https://doi.org/10.1109/CVPR.2017.233

[4] Karen Simonyan, Sander Dieleman, Andrew Senior, and Alex Graves. 2016. WaveNet. *arXiv Prepr. arXiv1609.03499v2* (2016). DOI:https://doi.org/10.1109/ICASSP.2009.4960364

[5] Tomas Mikolov, Wen Tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous spaceword representations. In NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference.

[6] Sebastian Ruder, Anders Søgaard, and Ivan Vulić. 2019. Unsupervised Cross-Lingual Representation Learning. DOI:https://doi.org/10.18653/v1/p19-4007

[7] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. DOI:https://doi.org/10.1109/CVPR.2015.7298594

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*.

[9] J. A. Sparano, R. J. Gray, D. F. Makower, K. I. Pritchard, K. S. Albain, D. F. Hayes, C. E. Geyer, E. C. Dees, M. P. Goetz, J. A. Olson, T. Lively, S. S. Badve, T. J. Saphner, L. I. Wagner, T. J. Whelan, M. J. Ellis, S. Paik, W. C. Wood, P. M. Ravdin, M. M. Keane, H. L. Gomez Moreno, P. S. Reddy, T. F. Goggins, I. A. Mayer, A. M. Brufsky, D. L. Toppmeyer, V. G. Kaklamani, J. L. Berenberg, J. Abrams, and G. W. Sledge. 2018. Adjuvant chemotherapy guided by a 21-gene expression assay in breast cancer. *N. Engl. J. Med.* (2018). DOI:https://doi.org/10.1056/NEJMoa1804710

[10] Mohamed Nooman Ahmed, Andeep S. Toor, Kelsey O'Neil, and Dawson Friedland. 2017. Cognitive Computing and the Future of Health Care Cognitive Computing and the Future of Healthcare: The Cognitive Power of IBM Watson Has the Potential to Transform Global Personalized Medicine. *IEEE Pulse* (2017). DOI:https://doi.org/10.1109/MPUL.2017.2678098

[11] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer. 2014. cudnn: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759, 2014.

[12] https://www.nvidia.com/en-gb/data-center/tesla-v100/

[13] TensorFlow: An open source machine learning framework for everyone. TensorFlow,[Online].Available: www.tensorflow.org/.

[14] PyTorch: An open source deep learning platform that provides a seamless path from research prototyping to production deployment. pytorch team. [Online]. Available: https://pytorch.org/

[15] Accelerating DNNs with Xilinx Alveo Accelerator Cards. *Xilinx* [Online]. Available: https://www.xilinx.com/

[16] Eriko Nurvitadhi, Ganesh Venkatesh, Jaewoong Sim, Debbie Marr, Randy Huang, Jason Gee Hock Ong, Yeong Tat Liew, Krishnan Srivatsan, Duncan Moss, Suchit Subhaschandra, and Guy Boudoukh. 2017. Can FPGAs beat GPUs in accelerating next-generation deep neural networks? In *FPGA 2017 - Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. DOI:https://doi.org/10.1145/3020078.3021740.

[17] Guo, K., Zeng, S., Yu, J., Wang, Y., & Yang, H. A Survey of FPGA-Based Neural Network Accelerator. arXiv 2017. *arXiv preprint arXiv:1712.08934*.

[18] Mike Davies, Narayan Srinivasa, Tsung Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, Yuyun Liao, Chit Kwan Lin, Andrew Lines, Ruokun Liu, Deepak Mathaikutty, Steven McCoy, Arnab Paul, Jonathan Tse, Guruguhanathan Venkataramanan, Yi Hsin Weng, Andreas Wild, Yoonseok Yang, and Hong Wang. 2018. Loihi: A Neuromorphic Manycore Processor with On-Chip Learning. *IEEE Micro* (2018). DOI:https://doi.org/10.1109/MM.2018.112130359

[19] Andrew S Cassidy, Jun Sawada, Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Filipp Akopyan, Bryan L Jackson, and Dharmendra S Modha. 2016. TrueNorth: a High-Performance, Low-Power Neurosynaptic Processor for Multi-Sensory Perception, Action, and Cognition. *IBM Res.* (2016).

[20] Goya Inference Platform and Performance Benchmarks. habana.ai,[Online]. Avalible: https://www.habana.ai/

[21] Ben Varkey Benjamin, Peiran Gao, Emmett McQuinn, Swadesh Choudhary, Anand R. Chandrasekaran, Jean Marie Bussat, Rodrigo Alvarez-Icaza, John V. Arthur, Paul A. Merolla, and Kwabena Boahen. 2014. Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations. *Proc. IEEE* (2014). DOI:https://doi.org/10.1109/JPROC.2014.2313565

[22] Johannes Schemmel, Johannes Fieres, and Karlheinz Meier. 2008. Wafer-scale integration of analog neural networks. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2008.4633828.

[23] Tesla Full Self Driving Chip design and architecture. Available online: https://en.wikichip.org/wiki/tesla_(car_company)/fsd_chip

[24] J. Joshua Yang, Dmitri B. Strukov, and Duncan R. Stewart. 2013. Memristive devices for computing. *Nature Nanotechnology*. DOI:https://doi.org/10.1038/nnano.2012.240

[25] Stuart A. Wolf, Almadena Y. Chtchelkanova, and Daryl M. Treger. 2006. Spintronics - A retrospective and perspective. *IBM Journal of Research and Development*. DOI:https://doi.org/10.1147/rd.501.0101

[26] Abhronil Sengupta and Kaushik Roy. 2017. Encoding neural and synaptic functionalities in electron spin: A pathway to efficient neuromorphic computing. *Applied Physics Reviews*. DOI:https://doi.org/10.1063/1.5012763

[27] Abu Sebastian, Manuel Le Gallo, Riduan Khaddam-Aljameh, and Evangelos Eleftheriou. 2020. Memory devices and applications for in-memory computing. *Nature Nanotechnology*. DOI:https://doi.org/10.1038/s41565-020-0655-z

[28] Shimeng Yu. 2018. Neuro-Inspired Computing with Emerging Nonvolatile Memorys. *Proc. IEEE* (2018). DOI:https://doi.org/10.1109/JPROC.2018.2790840

[29] Navnidhi K. Upadhyay, Hao Jiang, Zhongrui Wang, Shiva Asapu, Qiangfei Xia, and J. Joshua Yang. 2019. Emerging Memory Devices for Neuromorphic Computing. *Advanced Materials Technologies*. DOI:https://doi.org/10.1002/admt.201800589

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* (2017). DOI:https://doi.org/10.1145/3065386

[31] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. Cognitive modeling, 5(3), 1.

[32] Hebb, D.O. (1949). The Organization of Behavior. New York: Wiley & Sons.

[33] David E. Rumelhart and David Zipser. 1985. Feature discovery by competitive learning. *Cogn. Sci.* (1985). DOI:https://doi.org/10.1016/S0364-0213(85)80010-0

[34] Harel Z. Shouval, Samuel S.H. Wang, and Gayle M. Wittenberg. 2010. Spike timing dependent plasticity: A consequence of more fundamental learning rules. *Front. Comput. Neurosci.* (2010). DOI:https://doi.org/10.3389/fncom.2010.00019

[35] Yang, X. (2017). Understanding the variational lower bound.

[36] David Maxwell Chickering. 1996. Learning Bayesian Networks is NP-Complete. DOI:https://doi.org/10.1007/978-1-4612-2404-4_12

[37] Kullback, S.; Leibler, R. A. On Information and Sufficiency. Ann. Math. Statist. 22 (1951), no. 1, 79--86. doi:10.1214/aoms/1177729694. https://projecteuclid.org/euclid.aoms/1177729694

[38] David Maxwell Chickering, David Heckerman, and Christopher Meek. 2004. Large-sample learning of Bayesian networks is NP-hard. *J. Mach. Learn. Res.* (2004).

[39] Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: : networks of plausible inference*. DOI:https://doi.org/10.2307/2026705

[40] Stuart Geman and Donald Geman. 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans. Pattern Anal. Mach. Intell.* (1984). DOI:https://doi.org/10.1109/TPAMI.1984.4767596

[41] W. K. Hastings. 1970. Monte carlo sampling methods using Markov chains and their applications. *Biometrika* (1970). DOI:https://doi.org/10.1093/biomet/57.1.97

[42] J. Joshua Yang, Matthew D. Pickett, Xuema Li, Douglas A.A. Ohlberg, Duncan R. Stewart, and R. Stanley Williams. 2008. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nat. Nanotechnol.* (2008). DOI:https://doi.org/10.1038/nnano.2008.160

[43] Zhongrui Wang, Saumil Joshi, Sergey E. Savel'ev, Hao Jiang, Rivu Midya, Peng Lin, Miao Hu, Ning Ge, John Paul Strachan, Zhiyong Li, Qing Wu, Mark Barnell, Geng Lin Li, Huolin L. Xin, R. Stanley Williams, Qiangfei Xia, and J. Joshua Yang. 2017. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater.* (2017). DOI:https://doi.org/10.1038/nmat4756

[44] Pi, Shuang, Can Li, Hao Jiang, Weiwei Xia, Huolin Xin, J. Joshua Yang, and Qiangfei Xia. "Memristor crossbar arrays with 6-nm half-pitch and 2-nm critical dimension." Nature nanotechnology 14, no. 1 (2019): 35-39.

[45] Xiong, F., E. Yalon, A. Behnam, C. M. Neumann, K. L. Grosse, S. Deshmukh, and E. Pop. "Towards ultimate scaling limits of phase-change memory." In *2016 IEEE International Electron Devices Meeting (IEDM)*, pp. 4-1. IEEE, 2016.

[46] Tsai, Meng-Ju, Pin-Jui Chen, Dun-Bao Ruan, Fu-Ju Hou, Po-Yang Peng, Liu-Gu Chen, and Yung-Chun Wu. "Investigation of 5-nm-Thick Hf 0.5 Zr 0.5 O 2 ferroelectric FinFET dimensions for sub-60-mV/decade subthreshold slope." *IEEE Journal of the Electron Devices Society* 7 (2019): 1033-1037.

[47] H. S.Philip Wong, Simone Raoux, Sangbum Kim, Jiale Liang, John P. Reifenberg, Bipin Rajendran, Mehdi Asheghi, and Kenneth E. Goodson. 2010. Phase change memory. In *Proceedings of the IEEE*. DOI:https://doi.org/10.1109/JPROC.2010.2070050

[48] Zhengyang Zhao, Mahdi Jamali, Noel D'Souza, Delin Zhang, Supriyo Bandyopadhyay, Jayasimha Atulasimha, and Jian Ping Wang. 2016. Giant voltage manipulation of MgO-based magnetic tunnel junctions via localized anisotropic strain: A potential pathway to ultra-energy-efficient memory technology. *Appl. Phys. Lett.* (2016). DOI:https://doi.org/10.1063/1.4961670

[49] Park, J. 2020. Hybrid Non-Volatile Flip-Flops Using Spin-Orbit-Torque (SOT) Magnetic Tunnel Junction Devices for High Integration and Low Energy Power-Gating Applications. *MDPI Electronics.* (2020).

[50] Buhrman, R. 2003. Nano-processing and properties of spin transfer device structures. CC- 04. 10.1109/INTMAG.2003.1230327

[51] Yue Zhang, Weisheng Zhao, Yahya Lakys, Jacques Olivier Klein, Joo Von Kim, Dafiné Ravelosona, and Claude Chappert. 2012. Compact modeling of perpendicular-anisotropy CoFeB/MgO magnetic tunnel junctions. *IEEE Trans. Electron Devices* (2012). DOI:https://doi.org/10.1109/TED.2011.2178416

[52] Andrei Slavin and Vasil Tiberkevich. 2008. Excitation of spin waves by spin-polarized current in magnetic nano-structures. In *IEEE Transactions on Magnetics*. DOI:https://doi.org/10.1109/TMAG.2008.924537

[53] Dmitriy V. Dmitriev, Igor V. Marchishin, Andrey V. Goran, and Alexey A. Bykov. 2011. Microwave-induced giant oscillations of the magnetoconductivity and zero-conductance state in 2D electronic Corbino disks with capacitance contacts. In *12th International Conference and Seminar on Micro/Nanotechnologies and Electron Devices, EDM'2011 - Proceedings*. DOI:https://doi.org/10.1109/EDM.2011.6006956

[54] Yuansu Luo and Konrad Samwer. 2007. Oscillation of low-bias tunnel conductance with applied magnetic field in manganite/alumina tunnel structures. In *IEEE Transactions on Magnetics*. DOI:https://doi.org/10.1109/TMAG.2007.893692

[55] C. Mitsumata and A. Sakuma. 2011. Generalized model of antiferromagnetic domain wall. In *IEEE Transactions on Magnetics*. DOI:https://doi.org/10.1109/TMAG.2011.2158523

[56] Wang Kang, Chentian Zheng, Yangqi Huang, Xichao Zhang, Yan Zhou, Weifeng Lv, and Weisheng Zhao. 2016. Complementary Skyrmion Racetrack Memory with Voltage Manipulation. *IEEE Electron Device Lett.* (2016). DOI:https://doi.org/10.1109/LED.2016.2574916

[57] Fiore, A., Paranthoen, C., Chen, J. X., Ilegems, M., Mariucci, L., & Rossetti, M. 2003. Nanoscale quantum-dot light-emitting diodes. In *Quantum Electronics and Laser Science Conference* (p. QTuK2). Optical Society of America.

[58] Alessandro Spinelli and Andrea L. Lacaita. 1997. Physics and numerical simulation of single photon avalanche diodes. *IEEE Trans. Electron Devices* (1997). DOI:https://doi.org/10.1109/16.641363

[59] Hayden, O., Agarwal, R., & Lieber, C. M. (2006). Nanoscale avalanche photodiodes for highly sensitive and spatially resolved photon detection. *Nature Materials*. https://doi.org/10.1038/nmat1635

[60] Amirhasan Nourbakhsh, Ahmad Zubair, Redwan N. Sajjad, Tavakkoli K.G. Amir, Wei Chen, Shiang Fang, Xi Ling, Jing Kong, Mildred S. Dresselhaus, Efthimios Kaxiras, Karl K. Berggren, Dimitri Antoniadis, and Tomás Palacios. 2016. MoS2 Field-Effect Transistor with Sub-10 nm Channel Length. Nano Lett. (2016). DOI:https://doi.org/10.1021/acs.nanolett.6b03999

[61] Aaron D. Franklin, Mathieu Luisier, Shu Jen Han, George Tulevski, Chris M. Breslin, Lynne Gignac, Mark S. Lundstrom, and Wilfried Haensch. 2012. Sub-10 nm carbon nanotube transistor. Nano Lett. (2012). DOI:https://doi.org/10.1021/nl203701g

[62] Yunlong Guo, Gui Yu, and Yunqi Liu. 2010. Functional organic field-effect transistors. *Advanced Materials*. DOI:https://doi.org/10.1002/adma.201000740

[63] Jonathan Rivnay, Sahika Inal, Alberto Salleo, Róisín M. Owens, Magnus Berggren, and George G. Malliaras. 2018. Organic electrochemical transistors. *Nature Reviews Materials*. DOI:https://doi.org/10.1038/natrevmats.2017.86

[64] Mengwei Si, Pai Ying Liao, Gang Qiu, Yuqin Duan, and Peide D. Ye. 2018. Ferroelectric Field-Effect Transistors Based on MoS2 and CuInP2S6 Two-Dimensional van der Waals Heterostructure. *ACS Nano* (2018). DOI:https://doi.org/10.1021/acsnano.8b01810

[65] M. De Marchi, D. Sacchetto, S. Frache, J. Zhang, P. E. Gaillardon, Y. Leblebici, and G. De Micheli. 2012. Polarity control in double-gate, gate-all-around vertically stacked silicon nanowire FETs. In *Technical Digest - International Electron Devices Meeting, IEDM*. DOI:https://doi.org/10.1109/IEDM.2012.6479004

[66] Ren Li, Rawan Naous, Hossein Fariborzi, and Khaled Nabil Salama. 2019. Approximate Computing with Stochastic Transistors' Voltage Over-Scaling. *IEEE Access* (2019). DOI:https://doi.org/10.1109/ACCESS.2018.2889747

[67] Hyungjun Kim, Taesu Kim, Jinseok Kim, and Jae Joon Kim. 2018. Neural network optimized to resistive memory with nonlinear current-voltage characteristics. In *ACM Journal on Emerging Technologies in Computing Systems*. DOI:https://doi.org/10.1145/3145478

[68] Irina Kataeva, Farnood Merrikh-Bayat, Elham Zamanidoost, and Dmitri Strukov. 2015. Efficient training algorithms for neural networks based on memristive crossbar circuits. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2015.7280785

[69] Elham Zamanidoost, Michael Klachko, Dmitri Strukov, and Irina Kataeva. 2015. Low area overhead in-situ training approach for memristor-based classifier. In *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2015*. DOI:https://doi.org/10.1109/NANOARCH.2015.7180601

[70] Beiye Liu, Miao Hu, Hai Li, Zhi Hong Mao, Yiran Chen, Tingwen Huang, and Wei Zhang. 2013. Digital-assisted noise-eliminating training for memristor crossbar-based analog neuromorphic computing engine. In *Proceedings - Design Automation Conference*. DOI:https://doi.org/10.1145/2463209.2488741

[71] Damien Querlioz, Olivier Bichler, Philippe Dollfus, and Christian Gamrat. 2013. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Trans. Nanotechnol.* (2013). DOI:https://doi.org/10.1109/TNANO.2013.2250995

[72] Miao Hu, John Paul Strachan, Zhiyong Li, R. Stanley, and Williams. 2016. Dot-product engine as computing memory to accelerate machine learning algorithms. In *Proceedings - International Symposium on Quality Electronic Design, ISQED*. DOI:https://doi.org/10.1109/ISQED.2016.7479230

[73] Walt Woods and Christof Teuscher. 2017. Approximate vector matrix multiplication implementations for neuromorphic applications using memristive crossbars. In *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2017*. DOI:https://doi.org/10.1109/NANOARCH.2017.8053729

[74] Miguel Angel Lastras-Montano, Bhaswar Chakrabarti, Dmitri B. Strukov, and Kwang Ting Cheng. 2017. 3D-DPE: A 3D high-bandwidth dot-product engine for high-performance neuromorphic computing. In *Proceedings of the 2017 Design, Automation and Test in Europe, DATE 2017*. DOI:https://doi.org/10.23919/DATE.2017.7927183

[75] Miao Hu, Catherine E. Graves, Can Li, Yunning Li, Ning Ge, Eric Montgomery, Noraica Davila, Hao Jiang, R. Stanley Williams, J. Joshua Yang, Qiangfei Xia, and John Paul Strachan. 2018. Memristor-Based Analog Computation and Neural Network Classification with a Dot Product Engine. *Adv. Mater.* (2018). DOI:https://doi.org/10.1002/adma.201705914

[76] Miao Hu, John Paul Strachan, Zhiyong Li, Emmanuelle M. Grafals, Noraica Davila, Catherine Graves, Sity Lam, Ning Ge, Jianhua Joshua Yang, and R. Stanley Williams. 2016. Dot-product engine for neuromorphic computing: Programming 1T1M crossbar to accelerate matrix-vector multiplication. In *Proceedings - Design Automation Conference*. DOI:https://doi.org/10.1145/2897937.2898010

[77] Hussein Assaf, Yvon Savaria, and Mohamad Sawan. 2019. Memristor Emulators for an Adaptive DPE Algorithm: Comparative Study. In *Proceedings 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2019*. DOI:https://doi.org/10.1109/AICAS.2019.8771594

[78] Can Li, Miao Hu, Yunning Li, Hao Jiang, Ning Ge, Eric Montgomery, Jiaming Zhang, Wenhao Song, Noraica Dávila, Catherine E. Graves, Zhiyong Li, John Paul Strachan, Peng Lin, Zhongrui Wang, Mark Barnell, Qing Wu, R. Stanley Williams, J. Joshua Yang, and Qiangfei Xia. 2018. Analogue signal and image processing with large memristor crossbars. *Nat. Electron.* (2018). DOI:https://doi.org/10.1038/s41928-017-0002-z

[79] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov. 2015. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* (2015). DOI:https://doi.org/10.1038/nature14441

[80] Geoffrey W. Burr, Robert M. Shelby, Severin Sidler, Carmelo Di Nolfo, Junwoo Jang, Irem Boybat, Rohit S. Shenoy, Pritish Narayanan, Kumar Virwani, Emanuele U. Giacometti, Bulent N. Kurdi, and Hyunsang Hwang. 2015. Experimental Demonstration and Tolerancing of a Large-Scale Neural Network (165 000 Synapses) Using Phase-Change Memory as the Synaptic Weight Element. *IEEE Trans. Electron Devices* (2015). DOI:https://doi.org/10.1109/TED.2015.2439635

[81] F. Merrikh Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, and D. Strukov. 2018. Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits. *Nat. Commun.* (2018). DOI:https://doi.org/10.1038/s41467-018-04482-4

[82] Can Li, Zhongrui Wang, Mingyi Rao, Daniel Belkin, Wenhao Song, Hao Jiang, Peng Yan, Yunning Li, Peng Lin, Miao Hu, Ning Ge, John Paul Strachan, Mark Barnell, Qing Wu, R. Stanley Williams, J. Joshua Yang, and Qiangfei Xia. 2019. Long short-term memory networks in memristor crossbar arrays. *Nat. Mach. Intell.* (2019). DOI:https://doi.org/10.1038/s42256-018-0001-4

[83] Xinjie Guo, Farnood Merrikh-Bayat, Ligang Gao, Brian D. Hoskins, Fabien Alibart, Bernabe Linares-Barranco, Luke Theogarajan, Christof Teuscher, and Dmitri B. Strukov. 2015. Modeling and experimental demonstration of a hopfield network analog-to-digital converter with hybrid CMOS/memristor circuits. *Front. Neurosci.* (2015). DOI:https://doi.org/10.3389/fnins.2015.00488

[84] Can Li, Daniel Belkin, Yunning Li, Peng Yan, Miao Hu, Ning Ge, Hao Jiang, Eric Montgomery, Peng Lin, Zhongrui Wang, Wenhao Song, John Paul Strachan, Mark Barnell, Qing Wu, R. Stanley Williams, J. Joshua Yang, and Qiangfei Xia. 2018. Efficient and self-adaptive in-situ learning in

multilayer memristor neural networks. *Nat. Commun.* (2018). DOI:https://doi.org/10.1038/s41467-018-04484-2

[85] Chris Yakopcic, Raqibul Hasan, and Tarek M. Taha. 2015. Memristor based neuromorphic circuit for ex-situ training of multi-layer neural network algorithms. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2015.7280813

[86] Manan Suri, Vivek Parmar, Ashwani Kumar, Damien Querlioz, and Fabien Alibart. 2016. Neuromorphic hybrid RRAM-CMOS RBM architecture. In *2015 15th Non-Volatile Memory Technology Symposium, NVMTS 2015*. DOI:https://doi.org/10.1109/NVMTS.2015.7457484

[87] Amitesh Kumar, Mangal Das, Vivek Garg, Brajendra S. Sengar, Myo Than Htay, Shailendra Kumar, Abhinav Kranti, and Shaibal Mukherjee. 2017. Forming-free high-endurance Al/ZnO/Al memristor fabricated by dual ion beam sputtering. *Appl. Phys. Lett.* (2017). DOI:https://doi.org/10.1063/1.4989802

[88] D. Garbin, E. Vianello, O. Bichler, M. Azzaz, Q. Rafhay, P. Candelier, C. Gamrat, G. Ghibaudo, B. Desalvo, and L. Perniola. 2015. On the impact of OxRAM-based synapses variability on convolutional neural networks performance. In *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2015*. DOI:https://doi.org/10.1109/NANOARCH.2015.7180611

[89] Jeyavijayan Rajendran, Harika Maenm, Ramesh Karri, and Garrett S. Rose. 2011. An approach to tolerate process related variations in memristor-based applications. In *Proceedings of the IEEE International Conference on VLSI Design*. DOI:https://doi.org/10.1109/VLSID.2011.49

[90] Beiye Liu, Hai Li, Yiran Chen, Xin Li, Qing Wu, and Tingwen Huang. 2015. Vortex: Variation-aware training for memristor X-bar. In *Proceedings - Design Automation Conference*. DOI:https://doi.org/10.1145/2744769.2744930

[91] Jeyavijayan Rajendran, Ramesh Karri, and Garrett S. Rose. 2015. Improving tolerance to variations in memristor-based applications using parallel memristors. *IEEE Trans. Comput.* (2015). DOI:https://doi.org/10.1109/TC.2014.2308189

[92] Damien Querlioz, Olivier Bichler, and Christian Gamrat. 2011. Simulation of a memristor-based spiking neural network immune to device variations. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2011.6033439

[93] Sheng-Yang Sun, Zhiwei Li, Jiwei Li, Husheng Liu, Haijun Liu, and Qingjiang Li. 2019. A memristor-based convolutional neural network with full parallelization architecture. *IEICE Electronics Express*. 16-20181034.

[94] J. Joshua Yang, M. X. Zhang, Matthew D. Pickett, Feng Miao, John Paul Strachan, Wen Di Li, Wei Yi, Douglas A.A. Ohlberg, Byung Joon Choi, Wei Wu, Janice H. Nickel, Gilberto Medeiros-Ribeiro, and R. Stanley Williams. 2012. Engineering nonlinearity into memristors for passive crossbar applications. *Appl. Phys. Lett.* (2012). DOI:https://doi.org/10.1063/1.3693392

[95] Charles Augustine, Arijit Raychowdhury, Dinesh Somasekhar, James Tschanz, Vivek De, and Kaushik Roy. 2011. Design space exploration of typical STT MTJ stacks in memory arrays in the presence of variability and disturbances. *IEEE Trans. Electron Devices* (2011). DOI:https://doi.org/10.1109/TED.2011.2169962

[96] Aminul Islam, Mohd Ajmal Kafeel, Tanzeem Iqbal, and Mohd Hasan. 2012. Variability analysis of MTJ-based circuit. In *Proceedings of the 2012 3rd International Conference on Computer and Communication Technology, ICCCT 2012*. DOI:https://doi.org/10.1109/ICCCT.2012.20

[97] Jayita Das, Syed M. Alam, and Sanjukta Bhanja. 2012. Non-destructive variability tolerant differential read for non-volatile logic. In *Midwest Symposium on Circuits and Systems*. DOI:https://doi.org/10.1109/MWSCAS.2012.6291986

[98] Raffaele De Rose, Marco Lanuzza, Felice Crupi, Giulio Siracusano, Riccardo Tomasello, Giovanni Finocchio, and Mario Carpentieri. 2017. Variability-Aware Analysis of Hybrid MTJ/CMOS Circuits by a Micromagnetic-Based Simulation Framework. *IEEE Trans. Nanotechnol.* (2017). DOI:https://doi.org/10.1109/TNANO.2016.2641681

[99] Chris Yakopcic and Tarek M. Taha. 2013. Energy efficient perceptron pattern recognition using segmented memristor crossbar arrays. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2013.6707073

[100] Chris Yakopcic and Tarek M. Taha. 2013. Energy efficient perceptron pattern recognition using segmented memristor crossbar arrays. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2013.6707073

[101] Tarek M. Taha, Raqibul Hasan, and Chris Yakopcic. 2014. Memristor crossbar based multicore neuromorphic processors. In *International System on Chip Conference*. DOI:https://doi.org/10.1109/SOCC.2014.6948959

[102] Chris Yakopcic, Md Zahangir Alom, and Tarek M. Taha. 2017. Extremely parallel memristor crossbar architecture for convolutional neural network implementation. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2017.7966055

[103] Chris Yakopcic, Md Zahangir Alom, and Tarek M. Taha. 2016. Memristor crossbar deep network implementation based on a Convolutional neural network. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2016.7727302

[104] Yong Shim, Abhronil Sengupta, and Kaushik Roy. 2016. Low-power approximate convolution computing unit with domain-wall motion based “spin-memristor” for image processing applications. In *Proceedings - Design Automation Conference*. DOI:https://doi.org/10.1145/2897937.2898042

[105] Xiaoxiao Liu, Mengjie Mao, Beiye Liu, Hai Li, Yiran Chen, Boxun Li, Yu Wang, Hao Jiang, Mark Barnell, Qing Wu, and Jianhua Yang. 2015. RENO: A high-efficient reconfigurable neuromorphic computing accelerator design. In *Proceedings - Design Automation Conference*. DOI:https://doi.org/10.1145/2744769.2744900

[106] Yunji Chen, Tao Luo, Shaoli Liu, Shijin Zhang, Liqiang He, Jia Wang, Ling Li et al. 2014. Dadiannao: A machine-learning supercomputer. 47th Annual IEEE/ACM International Symposium on Microarchitecture, pp. 609-622.

[107] Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. 2016. ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars. In *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*. DOI:https://doi.org/10.1109/ISCA.2016.12

[108] Roman Kaplan, Leonid Yavits, and Ran Ginosar. 2018. PRINS: Processing-in-Storage Acceleration of Machine Learning. *IEEE Trans. Nanotechnol.* (2018). DOI:https://doi.org/10.1109/TNANO.2018.2799872

[109] Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. 2016. PRIME: A Novel Processing-in-Memory Architecture for Neural Network Computation in ReRAM-Based Main Memory. In *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*. DOI:https://doi.org/10.1109/ISCA.2016.13

[110] Cheng Xin Xue, Wei Hao Chen, Je Syu Liu, Jia Fang Li, Wei Yu Lin, Wei En Lin, Jing Hong Wang, Wei Chen Wei, Ting Wei Chang, Tung Cheng Chang, Tsung Yuan Huang, Hui Yao Kao, Shih Ying Wei, Yen Cheng Chiu, Chun Ying Lee, Chung Chuan Lo, Ya Chin King, Chorng Jung Lin, Ren Shuo Liu, Chih Cheng Hsieh, Kea Tiong Tang, and Meng Fan Chang. 2019. 24.1 A 1Mb Multibit ReRAM Computing-In-Memory Macro with 14.6ns Parallel MAC Computing Time for CNN Based AI Edge Processors. In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*. DOI:https://doi.org/10.1109/ISSCC.2019.8662395

[111] Qi Liu, Bin Gao, Peng Yao, Dong Wu, Junren Chen, Yachuan Pang, Wenqiang Zhang, Yan Liao, Cheng Xin Xue, Wei Hao Chen, Jianshi Tang, Yu Wang, Meng Fan Chang, He Qian, and Huaqiang Wu. 2020. A Fully Integrated Analog ReRAM Based 78.4TOPS/W Compute-In-Memory Chip with Fully Parallel MAC Computing. In *Digest of Technical Papers - IEEE International Solid-State Circuits Conference*. DOI:https://doi.org/10.1109/ISSCC19947.2020.9062953

[112] Lixue Xia, Boxun Li, Tianqi Tang, Peng Gu, Pai Yu Chen, Shimeng Yu, Yu Cao, Yu Wang, Yuan Xie, and Huazhong Yang. 2018. MNSIM: Simulation Platform for Memristor-Based Neuromorphic Computing System. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* (2018). DOI:https://doi.org/10.1109/TCAD.2017.2729466

[113] Xiaochen Peng, Shanshi Huang, Yandong Luo, Xiaoyu Sun, and Shimeng Yu. 2019. DNN+NeuroSim: An End-to-End Benchmarking Framework for Compute-in-Memory Accelerators with Versatile Device Technologies. In *Technical Digest - International Electron Devices Meeting, IEDM*. DOI:https://doi.org/10.1109/IEDM19573.2019.8993491

[114] Mazad S. Zaveri and Dan Hammerstrom. 2010. CMOL/CMOS Implementations of bayesian polytree inference: Digital and mixed-signal architectures and performance/price. *IEEE Trans. Nanotechnol.* (2010). DOI:https://doi.org/10.1109/TNANO.2009.2028342

[115] Santosh Khasanvis, Mingyu Li, Mostafizur Rahman, Ayan K. Biswas, Mohammad Salehi-Fashami, Jayasimha Atulasimha, Supriyo Bandyopadhyay, and Csaba Andras Moritz. 2015. Architecting for causal intelligence at nanoscale. *Computer (Long. Beach. Calif).* (2015). DOI:https://doi.org/10.1109/MC.2015.367

[116] Santosh Khasanvis, Mingyu Li, Mostafizur Rahman, Mohammad Salehi-Fashami, Ayan K. Biswas, Jayasimha Atulasimha, Supriyo Bandyopadhyay, and Csaba Andras Moritz. 2015. Self-Similar Magneto-Electric Nanocircuit Technology for Probabilistic Inference Engines. *IEEE Trans. Nanotechnol.* (2015). DOI:https://doi.org/10.1109/TNANO.2015.2439618

[117] Sourabh Kulkarni, Sachin Bhat, Santosh Khasanvis, and Csaba Andras Moritz. 2017. Magneto-electric approximate computational circuits for Bayesian inference. In *2017 IEEE International Conference on Rebooting Computing, ICRC 2017 - Proceedings*. DOI:https://doi.org/10.1109/ICRC.2017.8123678

[118] Xiaotao Jia, Jianlei Yang, Zhaohao Wang, Yiran Chen, Hai Helen Li, and Weisheng Zhao. 2018. Spintronics based stochastic computing for efficient Bayesian inference system. In *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*. DOI:https://doi.org/10.1109/ASPDAC.2018.8297385

[119] Sourabh Kulkarni, Sachin Bhat, and Csaba Andras Moritz. 2019. Reconfigurable probabilistic AI architecture for personalized cancer treatment. In *Proceedings of the 4th IEEE International Conference on Rebooting Computing, ICRC 2019*. DOI:https://doi.org/10.1109/ICRC.2019.8914697

[120] Z Kulesza, W Tylman. 2006. Implementation of Bayesian network in FPGA circuit. *ieeexplore.ieee.org*. Retrieved February 11, 2020 from https://ieeexplore.ieee.org/abstract/document/1706677/

[121] Mingjie Lin, Ilia Lebedev, and John Wawrzynek. 2010. High-throughput Bayesian computing machine with reconfigurable hardware. In *ACM/SIGDA International Symposium on Field Programmable Gate Arrays - FPGA*. DOI:https://doi.org/10.1145/1723112.1723127

[122] https://video-nvidia.com/en-gb/energy-nvidia-geforce

[123] Raqibul Hasan and Tarek M. Taha. 2014. Enabling back propagation training of memristor crossbar neuromorphic processors. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2014.6889893

[124] Djaafar Chabi, Zhaohao Wang, Christopher Bennett, Jacques Olivier Klein, and Weisheng Zhao. 2015. Ultrahigh Density Memristor Neural Crossbar for On-Chip Supervised Learning. *IEEE Trans. Nanotechnol.* (2015). DOI:https://doi.org/10.1109/TNANO.2015.2448554

[125] Manu V. Nair and Piotr Dudek. 2015. Gradient-descent-based learning in memristive crossbar arrays. In *Proceedings of the International Joint Conference on Neural Networks*. DOI:https://doi.org/10.1109/IJCNN.2015.7280658

[126] Daniel Soudry, Dotan Di Castro, Asaf Gal, Avinoam Kolodny, and Shahar Kvatinsky. 2015. Memristor-Based Multilayer Neural Networks With Online Gradient Descent Training. *IEEE Trans. Neural Networks Learn. Syst.* (2015). DOI:https://doi.org/10.1109/TNNLS.2014.2383395

[127] Cory Merkel and Dhireesha Kudithipudi. 2014. Neuromemristive extreme learning machines for pattern classification. In *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI*. DOI:https://doi.org/10.1109/ISVLSI.2014.67

[128] Stefano Ambrogio, Pritish Narayanan, Hsinyu Tsai, Robert M. Shelby, Irem Boybat, Carmelo Di Nolfo, Severin Sidler, Massimo Giordano, Martina Bodini, Nathan C.P. Farinha, Benjamin Killeen, Christina Cheng, Yassine Jaoudi, and Geoffrey W. Burr. 2018. Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature* (2018). DOI:https://doi.org/10.1038/s41586-018-0180-5

[129] Pai Yu Chen, Ligang Gao, and Shimeng Yu. 2016. Design of Resistive Synaptic Array for Implementing On-Chip Sparse Learning. *IEEE Trans. Multi-Scale Comput. Syst.* (2016). DOI:https://doi.org/10.1109/TMSCS.2016.2598742

[130] Boxun Li, Yuzhi Wang, Yu Weng, Yiran Chen, and Huazhong Yang. 2014. Training itself: Mixed-signal training acceleration for memristor-based neural network. In *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*. DOI:https://doi.org/10.1109/ASPDAC.2014.6742916

[131] Abhronil Sengupta, Aparajita Banerjee, and Kaushik Roy. 2016. Hybrid Spintronic-CMOS Spiking Neural Network with On-Chip Learning: Devices, Circuits, and Systems. *Phys. Rev. Appl.* (2016). DOI:https://doi.org/10.1103/PhysRevApplied.6.064003

[132] Yongtae Kim, Yong Zhang, and Peng Li. 2015. A reconfigurable digital neuromorphic processor with memristive synaptic crossbar for cognitive computing. *ACM J. Emerg. Technol. Comput. Syst.* (2015). DOI:https://doi.org/10.1145/2700234

[133] Christopher H. Bennett, Naimul Hassan, Xuan Hu, Jean Anne C. Incornvia, Joseph S. Friedman, and Matthew M. Marinella. 2019. Semi-supervised learning and inference in domain-wall magnetic tunnel junction (DW-MTJ) neural networks. DOI:https://doi.org/10.1117/12.2530308

[134] Mahdi Nazm Bojnordi and Engin Ipek. 2016. Memristive Boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. In *Proceedings - International Symposium on High-Performance Computer Architecture*. DOI:https://doi.org/10.1109/HPCA.2016.7446049

[135] Sukru Burc Eryilmaz, Emre Neftci, Siddharth Joshi, Sangbum Kim, Matthew Brightsky, Hsiang Lan Lung, Chung Lam, Gert Cauwenberghs, and Hon Sum Philip Wong. 2016. Training a Probabilistic Graphical Model with Resistive Switching Electronic Synapses. *IEEE Trans. Electron Devices* (2016). DOI:https://doi.org/10.1109/TED.2016.2616483

[136] Shamma Nasrin, Justine L. Drobitch, Supriyo Bandyopadhyay, and Amit Ranjan Trivedi. 2019. Low Power Restricted Boltzmann Machine Using Mixed-Mode Magneto-Tunneling Junctions. *IEEE Electron Device Lett.* (2019). DOI:https://doi.org/10.1109/LED.2018.2889881

[137] Johannes Bill and Robert Legenstein. 2014. A compound memristive synapse model for statistical learning through STDP in spiking neural networks. *Front. Neurosci.* (2014). DOI:https://doi.org/10.3389/fnins.2014.00412

[138] Bernhard Nessler, Michael Pfeiffer, Lars Buesing, and Wolfgang Maass. 2013. Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity. *PLoS Comput. Biol.* (2013). DOI:https://doi.org/10.1371/journal.pcbi.1003037

[139] Behtash Behin-Aein, Vinh Diep, and Supriyo Datta. 2016. A building block for hardware belief networks. *Sci. Rep.* (2016). DOI:https://doi.org/10.1038/srep29893

[140] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*.

[141] R. M. Neal. 1994. Bayesian Learning for Neural Networks. Ph.D. Thesis, Dept. of Computer Science, University of Toronto.

[142] Y. Kuramoto, and H. Araki. 1975. Lecture Notes in Physics. International Symposium on Mathematical Problems in Theoretical Physics(1975). 39. Springer-Verlag, New York. p. 420

[143] Dmitri E. Nikonov, Gyorgy Csaba, Wolfgang Porod, Tadashi Shibata, Danny Voils, Dan Hammerstrom, Ian A. Young, and George I. Bourianoff. 2015. Coupled-oscillator associative memory array operation for pattern recognition. *IEEE J. Explor. Solid-State Comput. Devices Circuits* (2015). DOI:https://doi.org/10.1109/JXCDC.2015.2504049

[144] Jacob Torrejon, Mathieu Riou, Flavio Abreu Araujo, Sumito Tsunegi, Guru Khalsa, Damien Querlioz, Paolo Bortolotti, Vincent Cros, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D. Stiles, and Julie Grollier. 2017. Neuromorphic computing with nanoscale spintronic oscillators. *Nature* (2017). DOI:https://doi.org/10.1038/nature23011

[145] Angeliki Pantazi, Stanisław Woźniak, Tomas Tuma, and Evangelos Eleftheriou. 2016. All-memristive neuromorphic computing with level-tuned neurons. *Nanotechnology* (2016). DOI:https://doi.org/10.1088/0957-4484/27/35/355205

[146] Abhronil Sengupta and Kaushik Roy. 2016. A Vision for All-Spin Neural Networks: A Device to System Perspective. *IEEE Trans. Circuits Syst. I Regul. Pap.* (2016). DOI:https://doi.org/10.1109/TCSI.2016.2615312

[147] Indranil Chakraborty, Gobinda Saha, Abhronil Sengupta, and Kaushik Roy. 2018. Toward Fast Neural Computing using All-Photonic Phase Change Spiking Neurons. *Sci. Rep.* (2018). DOI:https://doi.org/10.1038/s41598-018-31365-x

[148] Miguel Romera, Philippe Talatchian, Sumito Tsunegi, Flavio Abreu Araujo, Vincent Cros, Paolo Bortolotti, Juan Trastoy, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Maxence Ernoult, Damir Vodenicarevic, Tifenn Hirtzlin, Nicolas Locatelli, Damien Querlioz, and Julie Grollier. 2018. Vowel recognition with four coupled spin-torque nano-oscillators. *Nature*. DOI:https://doi.org/10.1038/s41586-018-0632-y

[149] Alexander Khitun, Guanxiong Liu, and Alexander A. Balandin. 2017. Two-dimensional oscillatory neural network based on room-temperature charge-density-wave devices. *IEEE Trans. Nanotechnol.* (2017). DOI:https://doi.org/10.1109/TNANO.2017.2716845

[150] Sourabh Kulkarni, Sachin Bhat, and Csaba Andras Moritz. 2017. Structure discovery for gene expression networks with emerging stochastic hardware. In *2017 IEEE International Conference on Rebooting Computing, ICRC 2017 - Proceedings*. DOI:https://doi.org/10.1109/ICRC.2017.8123688

[151] Brian Sutton, Kerem Yunus Camsari, Behtash Behin-Aein, and Supriyo Datta. 2017. Intrinsic optimization using stochastic nanomagnets. *Sci. Rep.* (2017). DOI:https://doi.org/10.1038/srep44370

[152] Naoya Onizawa, Daisaku Katagiri, Warren J. Gross, and Takahiro Hanyu. 2016. Analog-to-stochastic converter using magnetic tunnel junction devices for vision chips. *IEEE Trans. Nanotechnol.* (2016). DOI:https://doi.org/10.1109/TNANO.2015.2511151

[153] Rafatul Faria, Kerem Y. Camsari, and Supriyo Datta. 2018. Implementing Bayesian networks with embedded stochastic MRAM. *AIP Adv.* (2018). DOI:https://doi.org/10.1063/1.5021332

[154] Ramtin Zand, Kerem Yunus Camsari, Steven D. Pyle, Ibrahim Ahmed, Chris H. Kim, and Ronald F. DeMara. 2018. Low-Energy deep belief networks using intrinsic sigmoidal spintronic-based probabilistic neurons. In *Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI*. DOI:https://doi.org/10.1145/3194554.3194558

[155] Siyang Wang, Alvin R. Lebeck, and Chris Dwyer. 2015. Nanoscale Resonance Energy Transfer-Based Devices for Probabilistic Computing. *IEEE Micro* (2015). DOI:https://doi.org/10.1109/MM.2015.124

[156] Siyang Wang, Xiangyu Zhang, Yuxuan Li, Ramin Bashizade, Song Yang, Chris Dwyer, and Alvin R. Lebeck. 2016. Accelerating Markov Random Field Inference Using Molecular Optical Gibbs Sampling Units. In *Proceedings - 2016 43rd International Symposium on Computer Architecture, ISCA 2016*. DOI:https://doi.org/10.1109/ISCA.2016.55

[157] Xiangyu Zhang, Ramin Bashizade, Craig LaBoda, Chris Dwyer, and Alvin R. Lebeck. 2018. Architecting a stochastic computing unit with molecular optical devices. In *Proceedings - International Symposium on Computer Architecture*. DOI:https://doi.org/10.1109/ISCA.2018.00034

[158] Pierre Alexandre Blanche, Masoud Babaeian, Madeleine Glick, John Wissinger, Robert Norwood, Nasser Peyghambarian, Mark Neifeld, and Ratchaneekorn Thamvichai. 2016. Optical implementation of probabilistic graphical models. In *2016 IEEE International Conference on Rebooting Computing, ICRC 2016 - Conference Proceedings*. DOI:https://doi.org/10.1109/ICRC.2016.7738702

[159] Alice Mizrahi, Tifenn Hirtzlin, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Julie Grollier, and Damien Querlioz. 2018. Neural-like computing with populations of superparamagnetic basis functions. *Nat. Commun.* (2018). DOI:https://doi.org/10.1038/s41467-018-03963-w

[160] Sung Hyun Jo, Ting Chang, Idongesit Ebong, Bhavitavya B. Bhadviya, Pinaki Mazumder, and Wei Lu. 2010. Nanoscale memristor device as synapse in neuromorphic systems. *Nano Lett.* (2010). DOI:https://doi.org/10.1021/nl904092h

[161] Zhongrui Wang, Saumil Joshi, Sergey E. Savel'ev, Hao Jiang, Rivu Midya, Peng Lin, Miao Hu, Ning Ge, John Paul Strachan, Zhiyong Li, Qing Wu, Mark Barnell, Geng Lin Li, Huolin L. Xin, R. Stanley Williams, Qiangfei Xia, and J. Joshua Yang. 2017. Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing. *Nat. Mater.* (2017). DOI:https://doi.org/10.1038/nmat4756

[162] Duygu Kuzum, Rakesh G.D. Jeyasingh, Byoungil Lee, and H. S.Philip Wong. 2012. Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano Lett.* (2012). DOI:https://doi.org/10.1021/nl201040y

[163] Bryan L. Jackson, Bipin Rajendran, Gregory S. Corrado, Matthew Breitwisch, Geoffrey W. Burr, Roger Cheek, Kailash Gopalakrishnan, Simone Raoux, Charles T. Rettner, Alvaro Padilla, Alex G. Schrott, Rohit S. Shenoy, Bülent N. Kurdi, Chung H. Lam, and Dharmendra S. Modha. 2013. Nanoscale electronic synapses using phase change devices. *ACM J. Emerg. Technol. Comput. Syst.* (2013). DOI:https://doi.org/10.1145/2463585.2463588

[164] Michael L. Schneider, Christine A. Donnelly, Stephen E. Russek, Burm Baek, Matthew R. Pufall, Peter F. Hopkins, Paul D. Dresselhaus, Samuel P. Benz, and William H. Rippard. 2018. Ultralow power artificial synapses using nanotextured magnetic josephson junctions. *Sci. Adv.* (2018). DOI:https://doi.org/10.1126/sciadv.1701329

[165] Takeo Ohno, Tsuyoshi Hasegawa, Tohru Tsuruoka, Kazuya Terabe, James K. Gimzewski, and Masakazu Aono. 2011. Short-term plasticity and long-term potentiation mimicked in single inorganic synapses. *Nat. Mater.* (2011). DOI:https://doi.org/10.1038/nmat3054

[166] Abhronil Sengupta, Zubair Al Azim, Xuanyao Fong, and Kaushik Roy. 2015. Spin-orbit torque induced spike-timing dependent plasticity. *Appl. Phys. Lett.* (2015). DOI:https://doi.org/10.1063/1.4914111

[167] Matthew Jerry, Pai Yu Chen, Jianchi Zhang, Pankaj Sharma, Kai Ni, Shimeng Yu, and Suman Datta. 2018. Ferroelectric FET analog synapse for acceleration of deep neural network training. In *Technical Digest - International Electron Devices Meeting, IEDM*. DOI:https://doi.org/10.1109/IEDM.2017.8268338

[168] James M. Bower, David Beeman, Mark Nelson, and John Rinzel. 1995. The Hodgkin-Huxley Model. In *The Book of GENESIS*. DOI:https://doi.org/10.1007/978-1-4684-0189-9_4

[169] Doron Tal and Eric L. Schwartz. 1997. Computing with the leaky integrate-and-fire neuron: Logarithmic computation and multiplication. *Neural Comput.* (1997). DOI:https://doi.org/10.1162/neco.1997.9.2.305

[170] Arun V. Holden and Yin Shui Fan. 1992. From simple to simple bursting oscillatory behaviour via chaos in the Rose-Hindmarsh model for neuronal activity. *Chaos, Solitons and Fractals* (1992). DOI:https://doi.org/10.1016/0960-0779(92)90032-I

[171] Jack D. Cowan. 1990. Discussion: McCulloch-Pitts and related neural nets from 1943 to 1989. *Bull. Math. Biol.* (1990). DOI:https://doi.org/10.1007/BF02459569

[172] Matthew D. Pickett, Gilberto Medeiros-Ribeiro, and R. Stanley Williams. 2013. A scalable neuristor built with Mott memristors. *Nat. Mater.* (2013). DOI:https://doi.org/10.1038/nmat3510

[173] Tomas Tuma, Angeliki Pantazi, Manuel Le Gallo, Abu Sebastian, and Evangelos Eleftheriou. 2016. Stochastic phase-change neurons. *Nat. Nanotechnol.* (2016). DOI:https://doi.org/10.1038/nnano.2016.70

[174] Zhongrui Wang, Mingyi Rao, Jin Woo Han, Jiaming Zhang, Peng Lin, Yunning Li, Can Li, Wenhao Song, Shiva Asapu, Rivu Midya, Ye Zhuo, Hao Jiang, Jung Ho Yoon, Navnidhi Kumar Upadhyay, Saumil Joshi, Miao Hu, John Paul Strachan, Mark Barnell, Qing Wu, Huaqiang Wu, Qinru Qiu, R. Stanley Williams, Qiangfei Xia, and J. Joshua Yang. 2018. Capacitive neural network with neuro-transistors. *Nat. Commun.* (2018). DOI:https://doi.org/10.1038/s41467-018-05677-5

[175] Mrigank Sharad, Deliang Fan, and Kaushik Roy. 2013. Spin-neurons: A possible path to energy-efficient neuromorphic computers. *J. Appl. Phys.* (2013). DOI:https://doi.org/10.1063/1.4838096

[176] Wesley H. Brigner, Naimul Hassan, Xuan Hu, Lucian Jiang-Wei, Otitoaleke G. Akinola, Felipe Garcia-Sanchez, Massimo Pasquale, Christopher H. Bennett, Jean Anne C. Incorvia, and Joseph S. Friedman. 2019. Magnetic domain wall neuron with intrinsic leaking and lateral inhibition capability. DOI:https://doi.org/10.1117/12.2528218.

[177] Angeliki Pantazi, Stanisław Woźniak, Tomas Tuma, and Evangelos Eleftheriou. 2016. All-memristive neuromorphic computing with level-tuned neurons. *Nanotechnology* (2016). DOI:https://doi.org/10.1088/0957-4484/27/35/355205

[178] Abhronil Sengupta and Kaushik Roy. 2016. A Vision for All-Spin Neural Networks: A Device to System Perspective. *IEEE Trans. Circuits Syst. I Regul. Pap.* (2016). DOI:https://doi.org/10.1109/TCSI.2016.2615312

[179] Indranil Chakraborty, Gobinda Saha, Abhronil Sengupta, and Kaushik Roy. 2018. Toward Fast Neural Computing using All-Photonic Phase Change Spiking Neurons. *Sci. Rep.* (2018). DOI:https://doi.org/10.1038/s41598-018-31365-x

[180] D. B. Strukov, D. R. Stewart, J. Borghetti, X. Li, M. Pickett, G. Medeiros Ribeiro, W. Robinett, G. Snider, J. P. Strachan, W. Wu, Q. Xia, J. Joshua Yang, and R. S. Williams. 2010. Hybrid CMOS/memristor circuits. In *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems*. DOI:https://doi.org/10.1109/ISCAS.2010.5537020

[181] Kuk Hwan Kim, Siddharth Gaba, Dana Wheeler, Jose M. Cruz-Albrecht, Tahir Hussain, Narayan Srinivasa, and Wei Lu. 2012. A functional hybrid memristor crossbar-array/CMOS system for data storage and neuromorphic applications. *Nano Lett.* (2012). DOI:https://doi.org/10.1021/nl203687n

[182] Sachin Bhat, Sourabh Kulkarni, Jiajun Shi, Mingyu Li, and Csaba Andras Moritz. 2017. SkyNet: Memristor-based 3D IC for artificial neural networks. In *Proceedings of the IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2017*. DOI:https://doi.org/10.1109/NANOARCH.2017.8053706

[183] Shankar Ganesh Ramasubramanian, Rangharajan Venkatesan, Mrigank Sharad, Kaushik Roy, and Anand Raghunathan. 2015. SPINDLE: SPINtronic Deep Learning Engine for large-scale neuromorphic computing. In *Proceedings of the International Symposium on Low Power Electronics and Design*. DOI:https://doi.org/10.1145/2627369.2627625

[184] E. Covi, R. George, J. Frascaroli, S. Brivio, C. Mayr, H. Mostafa, G. Indiveri, and S. Spiga. 2018. Spike-driven threshold-based learning with memristive synapses and neuromorphic silicon neurons. *J. Phys. D. Appl. Phys.* 51, 34 (July 2018), 344003. DOI:https://doi.org/10.1088/1361-6463/aad361

[185] Ramtin Zand and Ronald F. DeMara. 2019. SNRA: A Spintronic Neuromorphic Reconfigurable Array for In-Circuit Training and Evaluation of Deep Belief Networks. In *2018 IEEE International Conference on Rebooting Computing, ICRC 2018*. DOI:https://doi.org/10.1109/ICRC.2018.8638604

[186] Jacob Torrejon, Mathieu Riou, Flavio Abreu Araujo, Sumito Tsunegi, Guru Khalsa, Damien Querlioz, Paolo Bortolotti, Vincent Cros, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D. Stiles, and Julie Grollier. 2017. Neuromorphic computing with nanoscale spintronic oscillators. *Nature* (2017). DOI:https://doi.org/10.1038/nature23011

[187] Mohammed A. Zidan, Yeon Joo Jeong, and Wei D. Lu. 2017. Temporal Learning Using Second-Order Memristors. *IEEE Trans. Nanotechnol.* (2017). DOI: https://doi.org/10.1109/TNANO.2017.2710158

[188] Mahyar Shahsavari, Pierre Falez, and Pierre Boulet. 2016. Combining a volatile and nonvolatile memristor in artificial synapse to improve learning in Spiking Neural Networks. In Proceedings of the 2016 IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2016. DOI: https://doi.org/10.1145/2950067.2950090

[189] Himanshu Thapliyal, Fazel Sharifi, and S. Dinesh Kumar. 2018. Energy-Efficient Design of Hybrid MTJ/CMOS and MTJ/Nanoelectronics Circuits. IEEE Trans. Magn. (2018). DOI: https://doi.org/10.1109/TMAG.2018.2833431

[190] Pi Feng Chiu, Meng Fan Chang, Che Wei Wu, Ching Hao Chuang, Shyh Shyuan Sheu, Yu Sheng Chen, and Ming Jinn Tsai. 2012. Low store energy, low VDDmin, 8T2R nonvolatile latch and SRAM with vertical-stacked resistive memory (memristor) devices for low power mobile applications. *IEEE J. Solid-State Circuits* (2012). DOI: https://doi.org/10.1109/JSSC.2012.2192661

[191] Said Hamdioui, Lei Xie, Hoang Anh Du Nguyen, Mottaqiallah Taouil, Koen Bertels, Henk Corporaal, Hailong Jiao, Francky Catthoor, Dirk Wouters, Linn Eike, and Jan Van Lunteren. 2015. Memristor based computation-in-memory architecture for data-intensive applications. In *Proceedings Design, Automation and Test in Europe, DATE*. DOI: https://doi.org/10.7873/date.2015.1136

[192] Alina Frolova and Bartek Wilczyński. 2018. Distributed Bayesian networks reconstruction on the whole genome scale. *PeerJ* (2018). DOI: https://doi.org/10.7717/peerj.5692

[193] Pournara, I., Bouganis, C. S., & Constantinides, G. A. (2005). FPGA-accelerated Bayesian learning for reconstruction of gene regulatory networks. *Proceedings - 2005 International Conference on Field Programmable Logic and Applications, FPL*. https://doi.org/10.1109/FPL.2005.1515742

[194] Ferreira, R., & Vendramini, J. C. G. (2010). FPGA-accelerated attractor computation of scale free gene regulatory networks. *Proceedings - 2010 International Conference on Field Programmable Logic and Applications, FPL 2010*. https://doi.org/10.1109/FPL.2010.108