© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722321

# Neuromorphic Computing

MANJULA BIRADAR
*Guest faculty, Department of computer science, Gulbarga University, kalburgi.*

***Abstract- The neuromorphic computing is a process that mimics the processing of information at biological synapses. The physical mechanism of neuromorphic computing is based on the ion transport at device interface to achieve efficient transfer and computation of information at the same location [11,12], which is different from logic circuit in digital systems [13,14]. Besides, neuromorphic computing is characterized by distributed memory and computational elements [15]. The neuromorphic computing can realize high-efficiency massive parallel computing that can conquer the limitation of Von Neumann bottleneck. Neuromorphic computing is a paradigm of designing hardware and algorithms inspired by the brain's architecture and principles, promising major gains in energy efficiency and new computing capabilities. This review provides a comprehensive overview of developments in neuromorphic computing from 2019 through 2024. We survey hardware advances – including digital neuromorphic chips (e.g. Intel Loihi, IBM TrueNorth, and SpiNNaker), emerging device technologies like memristors, spintronic circuits, photonic processors, and two-dimensional (2D) material-based devices – that enable brain-like computation with vastly lower power than conventional electronics. We also summarize algorithmic advances in spiking neural networks (SNNs), covering progress in temporal coding strategies, the introduction of surrogate gradient methods for training SNNs like deep networks, and biologically plausible learning rules such as e-prop for online learning in spiking systems. Furthermore, we discuss key opportunities and gaps: the potential of neuromorphic systems to approach aspects of human cognition or artificial general intelligence (AGI), applications in medicine (like brain–machine interfaces and neural prosthetics) and science, the trade-offs between power efficiency and computational precision, and challenges in integrating neuromorphic accelerators into existing computing ecosystems. We conclude by highlighting how co-development of hardware and algorithms is critical to fulfill the promise of neuromorphic computing, and by outlining open research directions on the path toward more brain-like, efficient computing architectures.***

***Index Terms: abstract, introduction, working, algorithm***

## I. INTRODUCTION

Neuromorphic computing, also known as neuromorphic engineering, is an approach to computing that mimics the way the human brain works. It entails designing hardware and software that simulate the neural and synaptic structures and functions of the brain to process information. Today, as artificial intelligence (AI) systems scale, they'll need state-of-the-art hardware and software behind them. Neuromorphic computing can act as a growth accelerator for AI, boost high-performance computing and serve as one of the building blocks of artificial superintelligence.

As a brain-inspired technology, neuromorphic computing also involves the notion of plasticity.
The human brain inspires a new class of computing architectures that radically depart from the conventional von Neumann paradigm. In a classical computer, memory and processing are separated, and operations occur in a sequential, clocked manner – a design that has led to tremendous performance gains but also faces power and scalability limits. Neuromorphic computing, first envisioned by Carver Mead in the 1980s, instead seeks to mimic the distributed, event-driven and parallel nature of brain networks. In a neuromorphic system, many simple processing units (artificial "neurons") operate in parallel and communicate via asynchronous spiking events, merging memory and computation locally at synapses. This design promises to circumvent the so-called von Neumann bottleneck (the limited bandwidth between processor and memory) by co-locating computation with memory, and to achieve dramatically higher energy efficiency akin to biological brains.

By 2019, neuromorphic computing had evolved from early analog circuits and small-scale prototypes into larger digital chips and emerging device technologies. Earlier milestones like IBM's

TrueNorth chip (2014) demonstrated a fully digital neurosynaptic processor with 1 million spiking neurons, running on only about 70 mW of power. Likewise, the SpiNNaker system (first phase completed ~2018) incorporated a million ARM cores to simulate spiking neural networks in real time, primarily aimed at large-scale brain simulations. These efforts showed that orders-of-magnitude gains in energy efficiency are possible; TrueNorth, for example, delivered ~46 billion synaptic operations per second per watt in some tasks. However, they also underscored challenges: TrueNorth's neural model was rigid and difficult to program for complex tasks, and many academic neuromorphic platforms lacked software support, limiting their practical use.
Since 2019, research in neuromorphic computing has accelerated along two broad fronts. Hardware developments have diversified beyond digital CMOS chips into analog/mixed-signal designs and exotic technologies (memristive devices, spintronic circuits, photonic and 2D-material-based neuromorphic devices). These new hardware platforms aim to improve scalability, density, and bio-realism of neural computations. At the same time, algorithmic advances in spiking neural networks have made it easier to perform useful computations on neuromorphic substrates. Notably, researchers developed methods to train SNNs with high accuracy through surrogate gradients, and explored learning rules that are more biologically plausible or hardware-friendly (such as local synaptic plasticity and reward-modulated learning). Neuromorphic algorithms have also expanded into applications like vision, sensory processing, robotics, and optimization, often leveraging the event-driven nature of SNNs for real-time processing of streaming data.

This article reviews the key hardware and algorithmic innovations in neuromorphic computing over 2019–2024, and discusses the emerging opportunities and remaining challenges. We begin by surveying the state-of-the-art neuromorphic hardware, from established digital chips to novel devices. We then cover progress in spiking neural network models, coding schemes, and learning algorithms that enable these systems to solve tasks. Finally, we examine how neuromorphic computing is being positioned in broader contexts – from efforts to approach brain-like intelligence, to use-cases in medicine and science – and identify gaps that must be addressed to realize the full potential of this paradigm.

Neuromorphic devices are designed for real-time learning, continuously adapting to evolving stimuli in the form of inputs and parameters. This means they could excel at solving novel problems.

## II. HOW NEUROMORPHIC COMPUTING WORKS

Since neuromorphic computing takes inspiration from the human brain, it borrows heavily from biology and neuroscience.

According to the Queensland Brain Institute, neurons "are the fundamental units of the brain and nervous system."5 As messengers, these nerve cells relay information between different areas of the brain and to other parts of the body. When a neuron becomes active or "spikes," it triggers the release of chemical and electrical signals that travel via a network of connection points called synapses, allowing neurons to communicate with each other.6

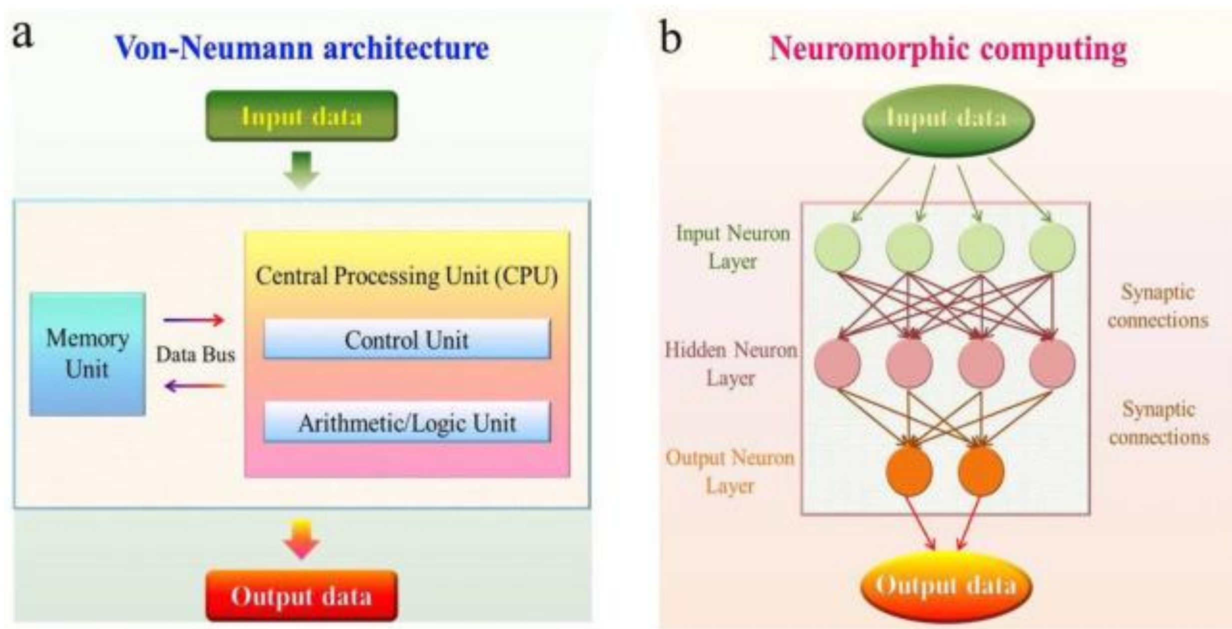
These neurological and biological mechanisms are modeled in neuromorphic computing systems through spiking neural networks (SNNs). A spiking neural network is a type of artificial neural network composed of spiking neurons and synapses.

Spiking neurons store and process data similar to biological neurons, with each neuron having its own charge, delay and threshold values. Synapses create pathways between neurons and also have delay and weight values associated with them. These values—neuron charges, neuron and synaptic delays, neuron thresholds and synaptic weights—can all be programmed within neuromorphic computing systems.7

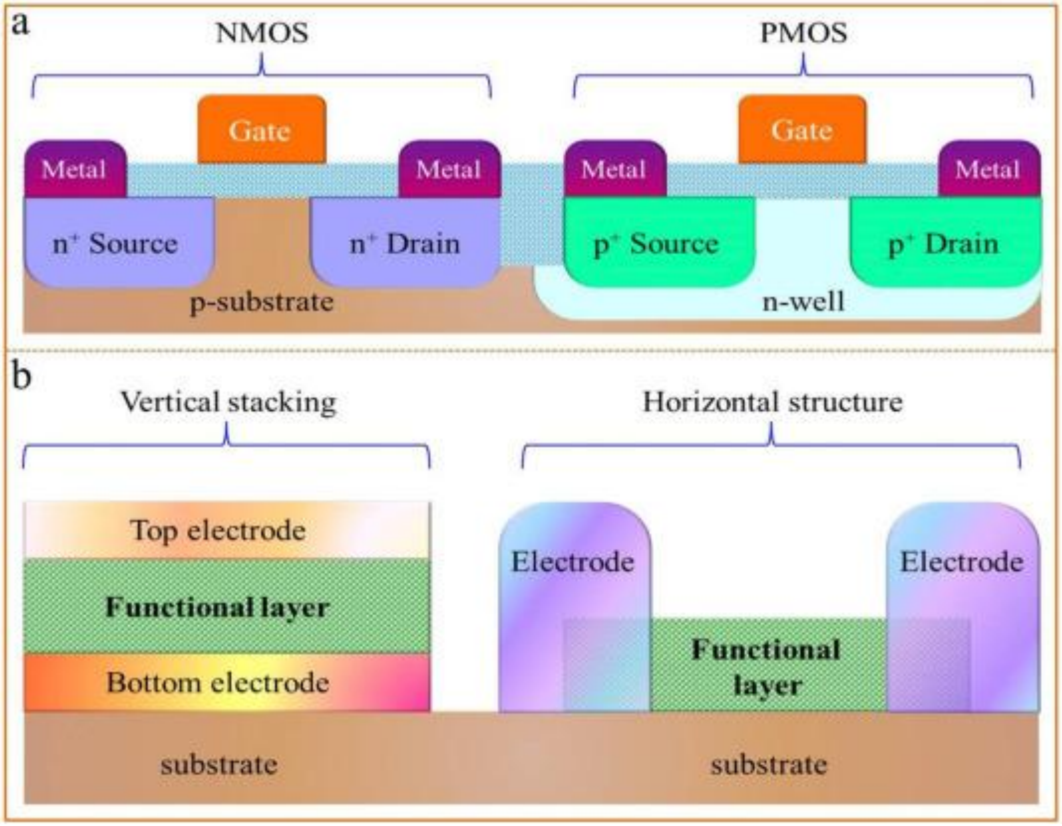
In neuromorphic architecture, synapses are represented as transistor-based synaptic devices, employing circuits to transmit electrical signals. Synapses typically include a learning component, altering their weight values over time according to activity within the spiking neural network.7

Unlike conventional neural networks, SNNs factor timing into their operation. A neuron's charge value accumulates over time; and when that charge reaches the neuron's associated threshold value, it spikes, propagating information along its synaptic web. But if the charge value doesn't go over the threshold, it dissipates and eventually "leaks." Additionally, SNNs are event-driven, with neuron and synaptic delay values allowing asynchronous dissemination of information.7

In 1990, Carver Mead firstly used "neuromorphic" to describe the transmission and processing of neural information with high computational efficiency, which is fundamentally different from digital computing [9]. As shown in Fig. 2, conventional Von Neumann architecture constraints the separated data memory and processing, while neuromorphic computing provides dual functionalities of data memory and computing at the same position without data transmission. This will result in less power consumption with nanojoules (nJ) and potentially higher execution speed with nanoseconds (ns) in neuromorphic computing than conventional Von Neumann architecture [10].




In order to implement neuromorphic computing, earlier attempts were predominantly focused on the transistor-based circuits. Farquhar et al. demonstrated a field programmable neural array based on floating gate transistors to model the channel-ion characteristics of biological neurons [16]. Further, Poon et al. demonstrated a chip that mimics the ion-based communication in a synapse between two neurons using analog circuits based on transistors [17]. The complementary metal oxide semiconductor (CMOS) based neuromorphic chip was designed [18]. The Truenorth chip only consumes the power of 1/10000th compared to conventional microprocessors [19], thus emphasizing significance of shifting towards the neuromorphic computing application. Although neuromorphic computing can be achieved using conventional CMOS devices, it takes considerable area to implement neuron functions. As shown in Fig. 3, CMOS devices are composed of n-metal-oxide-semiconductor (NMOS) and p-metal-oxide-semiconductor (PMOS). In comparison, the memristive devices, including vertical stacking and horizontal structure, provides a much simpler way to achieve that with easy fabrication, smaller size and high-packed density. In particular, the functional layer of memristors is not limited to semiconductor materials, and it can also use biomaterials and polymers [20]. The biomaterials based memristive devices not only provide a low manufacturing cost, but can also have the advantages of green, pollution-free and sustainability for neuromorphic devices.




III. NEUROMORPHIC COMPUTING ALGORITHMS:

In the software realm, developing training and learning algorithms for neuromorphic computing involves both machine learning and non-machine learning techniques. Here are a few of them:7

Deep learning
To perform inferencing, pretrained deep neural networks can be converted to spiking neural networks using mapping strategies such as normalizing weights or activation functions. A deep neural network can also be trained in a way that its neurons are activated like spiking neurons.

Evolutionary algorithms
These bio-inspired algorithms employ principles of biological evolution, such as mutation, reproduction and selection. Evolutionary algorithms can be used to design or train SNNs, changing and optimizing their parameters (delays and thresholds, for example) and structure (the number of neurons and the method of linking via synapses, for instance) as time passes.

Graphs
Spiking neural networks lend themselves well to a graph representation, with an SNN taking the form of a directed graph. When one of the nodes in the graph spikes, the time at which other nodes also spike coincides with how long the shortest path is from the originating node.

Plasticity
In neuroscience, neuroplasticity refers to the ability of the human brain and nervous system to modify its neural pathways and synapses in response to an injury. In neuromorphic architecture, synaptic plasticity is typically implemented through spike timing-dependent plasticity. This operation adjusts

© AUG 2026 | IRE Journals | Volume 10 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV10I2-1722321

the weights of synapses according to neurons' relative spike timings.

Reservoir computing
Reservoir computing, which is based on recurrent neural networks, uses a "reservoir" to cast inputs to a higher-dimension computational space, with a readout mechanism trained to read the reservoir's output.

In neuromorphic computing, input signals are fed to a spiking neural network, which acts as the reservoir. The SNN is untrained; instead, it relies on the recurrent connections within its network along with synaptic delays to map inputs to a higher-dimension computational space.

Benefits of neuromorphic computing
Neuromorphic systems hold a lot of computational promise. Here are some of the potential benefits this type of computing architecture offers:

Adaptability
As a brain-inspired technology, neuromorphic computing also involves the notion of plasticity. Neuromorphic devices are designed for real-time learning, continuously adapting to evolving stimuli in the form of inputs and parameters. This means they could excel at solving novel problems.

Energy efficiency
As mentioned previously, neuromorphic systems are event-based, with neurons and synapses processing in response to other spiking neurons. As a result, only the segment that's computing spikes consumes power while the rest of the network stays idle. This leads to more efficient energy consumption.

High performance
Most modern computers, also known as von Neumann computers, have separate central processing units and memory units and the transfer of data between these units can cause a bottleneck that impacts speed. On the other hand, neuromorphic computing systems both store and process data in individual neurons, resulting in lower latency and swifter computation compared to von Neumann architecture.

Parallel processing
Because of an SNN's asynchronous nature, individual neurons can perform different operations concurrently. So theoretically, neuromorphic devices can execute as many tasks as there are neurons at a given time. As such, neuromorphic architectures have immense parallel processing capabilities, allowing them to complete functions quickly.

Challenges of neuromorphic computing
Neuromorphic computing is still an emerging field. And like any technology in its early stages, neuromorphic systems face a few challenges:

Decreased accuracy
The process of converting deep neural networks to spiking neural networks can cause a drop in accuracy. Additionally, the memristors used in neuromorphic hardware may have cycle-to-cycle and device variations that can affect accuracy, as well as limits to synaptic weight values that can lower precision.7

Lack of benchmarks and standards
As a somewhat nascent technology, neuromorphic computing has a shortage of standards when it comes to architecture, hardware and software. Neuromorphic systems also don't have clearly defined and established benchmarks, sample datasets, testing tasks and metrics, so it becomes difficult to evaluate performance and prove effectiveness.

Limited accessibility and software
Most algorithmic approaches to neuromorphic computing still employ software designed for von Neumann hardware, which can confine the results to what von Neumann architecture can achieve. Meanwhile, APIs (application programming interfaces), coding models and programming languages for neuromorphic systems have yet to be developed or made more broadly available.

Steep learning curve
Neuromorphic computing is a complex domain, drawing from disciplines such as biology, computer science, electronic engineering, math, neuroscience and physics. This makes it difficult to comprehend outside of an academic lab specializing in neuromorphic research.

Use cases for neuromorphic computing

Current real-world applications for neuromorphic systems are sparse, but the computing paradigm can possibly be applied in these use cases:

Autonomous vehicles

Because of its high performance and orders of magnitude gains in energy efficiency, neuromorphic computing can help improve an autonomous vehicle's navigational skills, allowing for quicker course correction and improved collision avoidance while lowering energy emissions.

Cybersecurity

Neuromorphic systems can help detect unusual patterns or activity that could signify cyberattacks or breaches. And these threats can be thwarted rapidly owing to the low latency and swift computation of neuromorphic devices.

Edge AI

The characteristics of neuromorphic architecture make it suitable for edge AI. Its low power consumption can help with the short battery life of devices like smartphones and wearables, while its adaptability and event-driven nature fit the information processing methods of remote sensors, drones and other Internet of Things (IoT) devices.

Pattern recognition

Because of its extensive parallel processing capabilities, neuromorphic computing can be used in machine learning applications for recognizing patterns in natural language and speech, analyzing medical images and processing imaging signals from fMRI brain scans and electroencephalogram (EEG) tests that measure electrical activity in the brain.

Robotics

As an adaptable technology, neuromorphic computing can be used to enhance a robot's real-time learning and decision-making skills, helping it better recognize objects, navigate intricate factory layouts and operate faster in an assembly line.

Conclusion:

Neuromorphic computing successfully bridges biological principles and hardware design, offering breakthrough energy efficiency, low latency, and real-time adaptation by mimicking the human brain's neural structure.

## REFERENCES

[1] https://humanunsupervised.com/papers/neuromorphic_landscape.html

[2] https://www.sciencedirect.com/topics/materials-science/neuromorphic-computing

[3] https://www.ibm.com/think/topics/neuromorphic-computing

[4] Introduction to Neuromorphic Computing By Shriram Ramanathan and Abhronil Sengupta.

[5] Neuromorphic Computing: Architectures That Think Like Us VOL-I: Designing Brain-Inspired Systems for the Future of AI and Computing

[6] by Anshuman Mishra (Author)