Global Research Journal in Engineering, Science, Technology and Management (GRJESTM)
Volume 02, Issue 03 (May-June) 2026, pp. 387-392
GRJESTM :: An odaswa™ Journal
ISSN : 3107-3611 (Online)

TECHNICAL SERIES | TECHNOLOGY INSIGHT – TN12

Neuromorphic Computing and Spiking Neural Networks: Event-Driven Processing Beyond Conventional AI Hardware

Karan Pratap Singh[1]* and Meenakshi Rathore [2]

Technology Snapshot.

Technology Domain	AI systems and Information retrieval
Core Technologies	LLMs, vector search, knowledge graphs, hybrid retrieval, RAG
System Architecture	Retrieval + graph reasoning + generation
Primary Applications	Technical support, enterprise search, research assistance, knowledge management
Key Advantages	Higher context accuracy, better reasoning, reduced hallucinations
Key Challenges	Graph construction, scalability, latency, data freshness

Keywords: Retrieval-Augmented Generation, Knowledge Graphs, Graph-Based Retrieval, Vector Search, Large Language Models, Hybrid Retrieval, Grounded Generation.

* 1. Department of Computer Science and Engineering, Graphic Era University, Dehradun, Uttarakhand, India.
2. Department of Computer Science and Engineering, Amity University Rajasthan, Jaipur, Rajasthan, India.

Copyright: © 2026 Karan Pratap Singh and Meenakshi Rathore. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. Published By **odaswa**. Published in Chennai, Tamil Nadu, India

1. Introduction

Conventional computing architectures are primarily based on the von Neumann model, in which processing units and memory are organized as separate components. Data must frequently move between memory and processors during computation, creating communication overhead and energy consumption. This limitation becomes increasingly significant in artificial intelligence applications involving large neural networks and continuous data processing.

Neuromorphic computing introduces an alternative architecture inspired by the information-processing mechanisms of biological nervous systems. Instead of relying exclusively on conventional clock-driven operations and dense numerical computations, neuromorphic systems employ distributed processing, event-driven communication and neuron-like computational elements.

A major component of this approach is the Spiking Neural Network (SNN), which represents information through discrete electrical-like events called spikes. Unlike conventional artificial neural networks (ANNs), which commonly process continuous numerical activations, SNNs communicate through temporally distributed spike events. This enables computation based on both the occurrence and timing of signals.

Neuromorphic computing is attracting interest in applications requiring low-power processing, real-time response and continuous interaction with physical environments. Robotics, autonomous sensing, edge intelligence and event-based vision are among the potential application areas.

2. Technical Background of Spiking Neural Networks

Spiking Neural Networks are computational models inspired by biological neurons. A neuron receives signals from connected neurons through synaptic connections. These incoming signals influence its internal membrane potential. When the potential reaches a predefined threshold, the neuron generates a spike that propagates to connected neurons.

One commonly used computational model is the Leaky Integrate-and-Fire (LIF) neuron. In this model, incoming signals contribute to the membrane potential, while the potential gradually decreases over time due to leakage. When the threshold is reached, the neuron generates a spike and its membrane potential is reset.

The timing of spikes carries information. Two neurons may generate the same number of spikes but produce different computational outcomes depending on their firing patterns and temporal relationships. This temporal representation enables SNNs to process dynamic signals and time-dependent information.

Synaptic weights determine the influence of incoming spikes on a receiving neuron. Learning mechanisms modify these weights to improve the network's response to particular input patterns. Spike-Timing-Dependent Plasticity (STDP) is one such learning mechanism, in which synaptic strength is adjusted according to the relative timing of pre-synaptic and post-synaptic spikes.

SNNs therefore provide a computational framework in which neural activity, temporal dynamics and communication events are integrated into the learning and inference process.

3. Neuromorphic Computing Architecture

Neuromorphic hardware is designed to implement neural computation through specialized processing elements, local memory and communication networks. Rather than separating computation and data storage entirely, many neuromorphic architectures integrate synaptic storage and neuron processing within closely connected computational cores.

The architecture generally includes the following components:

Neuromorphic processing cores: These cores implement neuron models, synaptic operations and spike-generation mechanisms. Multiple cores

operate in parallel to support large neural networks.

Synaptic memory: Synaptic weights and connectivity information are stored close to processing elements. This reduces repeated data movement between separate memory and processing units.

Event-driven communication network: Neurons communicate through spike events rather than requiring every processing unit to exchange data continuously. Address-event representation (AER) is one established approach for transmitting event information between neuromorphic components.

Learning and adaptation mechanisms: Hardware or software modules implement learning rules that modify synaptic weights. Some platforms support programmable learning mechanisms directly within the neuromorphic processor.

Input and output interfaces: Sensors and application processors convert physical or digital inputs into spike representations and interpret the network's output for subsequent operations.

The resulting architecture supports distributed computation and sparse communication. When only a small proportion of neurons are active at a given time, unnecessary computations can be reduced.

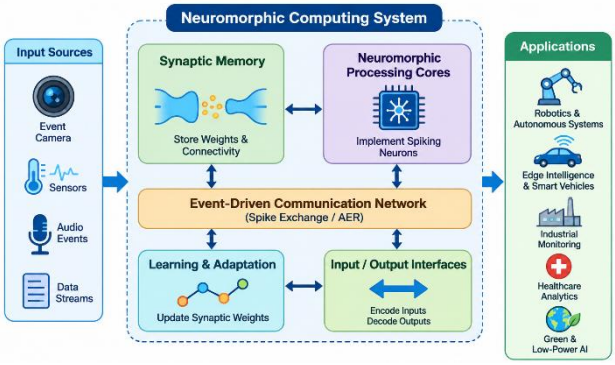

Fig. 1. Conceptual architecture of a neuromorphic computing system showing neuron processing cores, synaptic memory, event-driven communication and external sensor interfaces.

4. Event-Driven Processing and Computational Efficiency

A defining characteristic of neuromorphic computing is event-driven processing. In conventional digital architectures, processors generally operate according to clock cycles, even when individual computational units have little useful work to perform. Neuromorphic systems can instead activate processing operations in response to incoming spike events.

This approach offers several potential advantages.

Sparse computation: Only active neurons and relevant synaptic connections need to participate in a particular processing event. This can reduce unnecessary operations when input activity is sparse.

Reduced data movement: Local storage of synaptic information minimizes communication between separate memory and processing units.

Temporal information processing: Spike timing provides a mechanism for representing time-dependent patterns, making SNNs suitable for dynamic sensory signals.

Parallel processing: Multiple neuron groups can process incoming events concurrently, supporting distributed computational workloads.

Low-power operation: Event-driven activity and specialized hardware can reduce energy consumption for suitable workloads, particularly continuous sensing and edge inference.

However, these advantages are workload-dependent. Dense spike activity, inefficient data conversion, network communication overhead and unsuitable model implementations can reduce the expected energy benefits. Neuromorphic processors should therefore be evaluated against conventional hardware using equivalent tasks, accuracy requirements and system-level energy measurements.

5. Neuromorphic Hardware Platforms

Several research and development platforms demonstrate different approaches to implementing neuromorphic computation.

Intel Loihi and Loihi 2: Intel's Loihi architecture uses specialized neuromorphic cores connected through an on-chip communication network. Loihi 2 extends the architecture with more flexible neuron models, improved programmability and support for more advanced spike-based computation. These processors are designed for research involving SNNs, adaptive learning and event-driven applications.

IBM TrueNorth: TrueNorth is a digital neurosynaptic processor developed to support large-scale, low-power neural computation. Its architecture uses specialized neurosynaptic cores and event-driven communication. It demonstrates how large networks of simplified neuron models can be implemented using dedicated hardware.

SpiNNaker and SpiNNaker 2: Developed at the University of Manchester, SpiNNaker uses a massively parallel processor architecture to simulate and execute spiking neural networks. Its distributed processing model supports large-scale neural simulations and real-time interaction with external systems.

BrainScaleS-2: This platform uses mixed-signal neuromorphic hardware to emulate neural and synaptic dynamics. It is designed for accelerated neural computation and research into biologically inspired learning mechanisms.

These platforms demonstrate that neuromorphic computing is not restricted to a single hardware design. Digital, analogue and mixed-signal architectures provide different balances between flexibility, scalability, precision and energy efficiency.

6. Applications of Neuromorphic Computing

Neuromorphic computing has potential applications in several areas where continuous data processing, rapid response and energy efficiency are important.

Robotics and autonomous systems: SNNs can process sensory events and support real-time control decisions. Their temporal processing capabilities are relevant to robotic navigation, obstacle detection and adaptive motor control.

Event-based vision: Event cameras generate asynchronous visual events in response to changes in brightness rather than capturing complete image frames at fixed intervals. Neuromorphic processors can process these events directly, supporting motion detection and high-speed visual analysis.

Edge artificial intelligence: Battery-powered devices and embedded systems can use event-driven neural processing for continuous monitoring, anomaly detection and sensor-based classification.

Industrial monitoring: Neuromorphic systems can analyse time-dependent sensor signals to identify abnormal operating conditions and changing equipment behaviour.

Biomedical applications: Neural signal processing, brain-inspired modelling and wearable sensing are potential application areas where temporal information and low-power computation are important.

Autonomous sensing: Neuromorphic architectures can support always-on monitoring systems that respond selectively to significant changes in their environment.

7. Challenges and Research Directions

Despite its potential, neuromorphic computing faces several technical and practical challenges.

Training complexity: Conventional neural networks benefit from mature training methods such as backpropagation. Training deep SNNs is more complicated because spike-generation functions are non-differentiable. Surrogate-gradient methods, temporal learning techniques and specialized training algorithms are being investigated to address this limitation.

Hardware and software interoperability: Different neuromorphic platforms implement

different neuron models, communication mechanisms and programming interfaces. This makes it difficult to transfer trained networks between hardware systems without modification.

Limited software ecosystem: Compared with conventional GPU-based deep learning, neuromorphic development has a less mature ecosystem of tools, libraries and deployment frameworks.

Benchmarking difficulties: Energy efficiency depends on network architecture, spike activity, hardware implementation, workload and data-conversion overhead. Standardized benchmarking is required to establish meaningful comparisons with CPUs, GPUs and other AI accelerators.

Scalability: Large-scale neuromorphic systems require efficient communication networks, memory management and synchronization strategies. Inter-chip communication can become a significant limitation as network size increases.

Application suitability: SNNs are not automatically more efficient or accurate than conventional neural networks. Their advantages are most relevant to workloads that can exploit sparse, temporal and event-driven computation.

Future development is expected to focus on scalable neuromorphic processors, improved learning algorithms, hardware-software co-design and integration with event-based sensors. Research into hybrid ANN-SNN architectures may also enable conventional neural-network models and spiking computation to operate together.

8. Conclusion

Neuromorphic computing represents an alternative approach to conventional processor-based artificial intelligence through brain-inspired architectures and event-driven computation. Spiking Neural Networks provide a framework for processing information through discrete temporal events, while specialized hardware integrates neuron computation, synaptic memory and communication mechanisms. Platforms such as Intel Loihi 2, SpiNNaker 2 and BrainScaleS-2 demonstrate the growing capabilities of this computing paradigm. Although challenges remain in training, interoperability, scalability and application deployment, neuromorphic computing offers promising opportunities for low-power edge intelligence, autonomous sensing and real-time robotic systems.

References

[1] C. Mead, "Neuromorphic Electronic Systems," Proceedings of the IEEE, Volume 78, Issue 10, pp. 1629–1636, 1990.

[2] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang and H. Wang, "Loihi: A Neuromorphic Manycore Processor with On-Chip Learning," IEEE Micro, Volume 38, Issue 1, pp. 82–99, 2018.

[3] S. B. Furber, F. Galluppi, S. Temple and L. Plana, "The SpiNNaker Project," Proceedings of the IEEE, Volume 102, Issue 5, pp. 652–665, 2014.

[4] A. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J. Kuang, R. Manohar, W. P. Risk, B. Jackson and D. S. Modha, "TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 34, Issue 10, pp. 1537–1557, 2015.

[5] A. Gebregiorgis, A. Yousefzadeh, S. Eissa, M. A. Siddiqi, C. Frenkel, F. Zenke, S. Bohte, A. N. Mahmoud, A. Das, S. Hamdioui, H. Corporaal and F. Corradi, "Spike-Based Neuromorphic Computing: An Overview from Bio-Inspiration to Hardware Architectures and Learning Mechanisms," Microprocessors and Microsystems, Volume 124, Part A, 105240, 2025.

[6] G. Indiveri and S.-C. Liu, "Memory and Information Processing in Neuromorphic Systems," Proceedings of the IEEE, Volume 103, Issue 8, pp. 1379–1397, 2015.

[7] D. Kudithipudi., C. Schuman, C.M. Vineyard, T. Pandit, C. Merkel, R. Kubendran, J.B Aimone, G. Orchard, C. Mayr, Benosman, R., Hays, J., Young, C., Bartolozzi, C., Majumdar, A., Cardwell, S.G., Payvand, M., Buckley, S., Kulkarni, S., Gonzalez, H.A., Cauwenberghs, G., Thakur, C.S., Subramoney, A. and Furber, S.. Neuromorphic computing at scale. *Nature*, [online] 637(8047), pp.801–812, 2025.