Introduction

Artificial intelligence is a transformative branch of computer science that focuses on creating systems capable of performing tasks that typically require human intelligence, encompassing abilities such as learning, reasoning, problem-solving, perception, and language understanding.

Although the sector has witnessed significant growth since its inception, its development faces multiple challenges, such as:

- Computational power limitations,
- Data transfer bottlenecks in von Neumann architecture,
- Scalability issues.

Neuromorphic computing mitigates these issues.

This whitepaper discusses neuromorphic computing, how its principles can be integrated into artificial intelligence, and introduces the **Neuro AI Agent** - an artificial intelligence system that utilizes neuromorphic computing.

Low computational power

Advanced AI algorithms require vast computational power for optimal performance. Modern deep learning and machine learning algorithms are increasingly requiring more cores, GPUs, and memory. Operating AI-driven applications in existing data centers or cloud environments will become progressively more difficult.

The demand for extremely high computational power by modern AI algorithms significantly hinders development of advanced artificial intelligence systems. For organizations to implement advanced artificial intelligence systems using conventional deep learning and machine learning algorithms, they must develop machines with computational power equivalent to supercomputers. Creating these machines is extremely expensive.

Data transfer bottlenecks

Modern AI algorithms are based on von Neumann architecture. One of the distinctive features of the von Neumann architecture is the separation of the processing unit and the memory. The processing unit is specifically designed to process data according to program instructions, while the memory is designated for holding data.

Since the processing unit is separated from memory, data must be transferred between the two components to facilitate operations. This design is susceptible to various issues.

The need to frequently transfer data between different components can slow down processing speed, hence reducing performance.

Frequent data transfers between the processing unit and memory are vulnerable to errors that may stop the execution of programs. This can hamper proper functioning of automated systems.

The von Neumann architecture is not well suited for executing simultaneous processes. The processing unit is designed to handle only one process at a time, switching between programs.

Energy used to facilitate data transfers between processing units and memory makes the model less energy-efficient.

Limited scalability

One of the fundamental requirements of automated systems is the ability to scale with emerging needs and evolve with changing technology. While the current architecture supports the expansion of systems without making significant changes, it is not well designed to support frequent adjustments that characterize artificial intelligence systems.

AI systems are in a constant state of adaptation. As new data and capabilities are consistently integrated, these systems need to constantly adopt new execution models. The von Neumann architecture is not designed to autonomously change as the system learns. Besides, adding new infrastructure to these systems can be challenging.

The von Neumann computing architecture is not designed with scalability at its core, and it is poorly equipped to support the learning capabilities of artificial intelligence systems. It has become imperative to explore alternative computing architectures.

Neuromorphic computing is one of the architectures that can unlock the true potential of artificial intelligence.

Neuromorphic Computing

A primary objective in artificial intelligence research is to create computers capable of learning and reasoning similar to human cognition. While there are various approaches to achieve this, the consensus within the engineering community is that the most effective approach involves creating computer models that replicate the human brain's architecture.

Neuromorphic computing is a process that mimics the human brain's structure and functionality, using artificial neurons and synapses to process information.

Using artificial neurons and synapses, neuromorphic models simulate how our brain processes information, allowing them to solve problems, recognize patterns, and make decisions more quickly and efficiently than conventional computing models. A **Neuron**, also called a node, is a basic computational unit that processes inputs and produces an output, using a weighted sum and an activation function. A **Synapse** is a connection between two neurons.

Unlike the von Neumann model, where processing units and memory are separate, the neuromorphic computing model integrates memory and processing units in the neurons and synapses. Neuromorphic algorithms are defined by the structure of the neural network and its parameters rather than by direct instructions, as in von Neumann architecture.

Another significant distinction is how input data is processed.

Instead of encoding information as numerical values in binary format, neuromorphic computing uses **Spikes** as inputs, where the timing, magnitude, and shape of these spikes encode numerical information.

Principles of Neuromorphic Computing

Parallel Processing

Neuromorphic computing inherently operates on a parallel structure. In this system, every neuron and synapse functions autonomously and simultaneously, working alongside others. This independence and simultaneous operation of neurons and synapses significantly boost the computational capacity of neuromorphic systems. By distributing tasks among many neurons and synapses, it avoids overburdening any single processing unit with too many tasks.

Unified Memory and Processing

As previously mentioned, in neuromorphic computing, memory and processing functions are unified within a single component. While neurons are generally viewed as processing units and synapses as storage units, in many neuromorphic systems, both neurons and synapses are responsible for data processing and storage tasks. This unification helps to overcome von Neumann bottlenecks. Additionally, combining memory and processing into one element also contributes to reducing power consumption.

Scalability

Neuromorphic systems possess an intrinsic ability to scale. Enhancing these systems with new capabilities or functions merely involves adding more neurons and synapses. Moreover, upgrading neuromorphic systems can be straightforwardly achieved by programming additional

virtual neurons and synapses. Such scalability is a crucial aspect, especially for ever-growing networks like artificial intelligence systems.

Event-Driven Computation

Neuromorphic systems utilize an event-driven computational approach. This means that these systems become active only when there is a need for computation and the necessary data is present. Neurons and synapses in these systems are engaged solely when there are spikes, or data events, to process. Generally, such spikes are infrequent in the network's operation. The capacity of neuromorphic systems to remain inactive during periods of no workload significantly lowers their energy requirements.

Stochasticity

Neuromorphic systems

Integrating Neuromorphic Computing with Artificial Intelligence

The principles of neuromorphic computing align effectively with the characteristics and requirements of artificial intelligence systems. They not only boost the processing capabilities of AI but also address challenges like the high power consumption of AI systems and data transfer bottlenecks. Integrating neuromorphic computing with artificial intelligence is mainly achieved by employing neuromorphic algorithms, which include:

- Spiking Neural Networks (SNN),
- Advanced algorithms.

Spiking Neural Networks (SNN)

This group of algorithms trains artificial intelligence systems by tuning the states and parameters of artificial neurons and synapses, facilitating learning new behavior by achieving new homeostasis.

SNN algorithms leverage the plasticity nature of ANN systems.

Plasticity is the ability of a neural network to quickly change its predictions in response to new information. It is essential for the adaptability and robustness of artificial intelligence systems.

We utilize the following algorithms within the SNN framework:

- **Spike-Timing-Dependent Plasticity (STDP):** STDP represents the most conventional form of SNN training. In this approach, the learning process is driven by the timing of

spikes between two interconnected neurons. As an unsupervised model, STDP is particularly well-suited for artificial intelligence systems where precision is not the primary focus.

Backpropagation-based direct training schemes: This is one of the most applied learning algorithms in artificial intelligence systems. The algorithm tests for errors working back from output nodes to input nodes. These methods are considered among the most effective approaches for training networks because of their high accuracy.

Supervised temporal learning: In these models, the classification process depends on the firing times of output neurons. Being supervised models, they tend to be more precise than STDP. However, these models are more suitable for binary classification, as they are less effective for multi-class classification tasks.

ANN-to-SNN conversion strategies: A network is first trained in the ANN framework, and the trained network is then converted into an SNN. Though the conversion process results in a slight loss, this approach is extremely beneficial because ANN training schemes are very mature and yield high accuracy. Moreover, this approach is suitable for ultra-large networks since it is not as complex as direct training.

Advanced Algorithms

Additional algorithms for integrating neuromorphic computing principles into advanced artificial intelligence systems include:

Reservoir computing (LSM),
Genetic algorithms.

Reservoir computing is characterized by the use of a sparsely interconnected Spiking Neural Network (SNN), serving as the reservoir. This reservoir is randomly structured but maintains two essential properties: input separability, ensuring different inputs lead to different outputs, and fading memory, which prevents signals from propagating indefinitely, causing them to diminish over time. Additionally, reservoir computing incorporates a readout mechanism like linear regression, trained to interpret the output from the reservoir. A primary benefit of this approach is that it eliminates the need for training the SNN component.

Genetic algorithms constantly adjust the system using available data to carry out new functions. These methods refine artificial intelligence systems by modifying their parameters, neurons, and synaptic thresholds. The appeal of these models lies in their versatility, as they can be

implemented in any network structure. Additionally, they enable the easy transformation of artificial intelligence networks over time.