computers

MDPI

Article

Implementing Tensor-Organized Memory for Message Retrieval Purposes in Neuromorphic Chips

Arash Khajooei Nejad [1], Mohammad (Behdad) Jamshidi [2,*] and Shahriar B. Shokouhi [1]

[1] School of Electrical Engineering, Iran University of Science and Technology, Tehran 13114-16846, Iran
[2] The International Association of Engineers, 37-39 Hung To Road, Hong Kong 999077, China
* Correspondence: bmj.jmd@gmail.com

Abstract: This paper introduces Tensor-Organized Memory (TOM), a novel neuromorphic architecture inspired by the human brain's structural and functional principles. Utilizing spike-timing-dependent plasticity (STDP) and Hebbian rules, TOM exhibits cognitive behaviors similar to the human brain. Compared to conventional architectures using a simplified leaky integrate-and-fire (LIF) neuron model, TOM showcases robust performance, even in noisy conditions. TOM's adaptability and unique organizational structure, rooted in the Columnar-Organized Memory (COM) framework, position it as a transformative digital memory processing solution. Innovative neural architecture, advanced recognition mechanisms, and integration of synaptic plasticity rules enhance TOM's cognitive capabilities. We have compared the TOM architecture with a conventional floating-point architecture, using a simplified LIF neuron model. We also implemented tests with varying noise levels and partially erased messages to evaluate its robustness. Despite the slight degradation in performance with noisy messages beyond 30%, the TOM architecture exhibited appreciable performance under less-than-ideal conditions. This exploration into the TOM architecture reveals its potential as a framework for future neuromorphic systems. This study lays the groundwork for future applications in implementing neuromorphic chips for high-performance intelligent edge devices, thereby revolutionizing industries and enhancing user experiences within the power of artificial intelligence.

Keywords: neuromorphic engineering; Tensor-Organized Memory (TOM); spike-timing-dependent plasticity (STDP); Hebbian rules; leaky integrate-and-fire (LIF) neuron model; spiking neural networks; memory storage and retrieval; noise tolerance; neuromorphic systems; intelligent systems

Citation: Khajooei Nejad, A.; Jamshidi, M.; B. Shokouhi, S. Implementing Tensor-Organized Memory for Message Retrieval Purposes in Neuromorphic Chips. *Computers* **2023**, *12*, 189. https://doi.org/10.3390/computers12100189

Academic Editor: Paolo Bellavista

Received: 8 August 2023
Revised: 16 September 2023
Accepted: 18 September 2023
Published: 22 September 2023

1. Introduction

In addition to the advancements in artificial intelligence and hardware technologies, the field of neuromorphic engineering presents a promising avenue for unlocking the full potential of brain-inspired architecture chips. By leveraging insights from neurobiology and replicating the brain's structure and organization in artificial systems, neuromorphic engineering enables the development of novel computing architectures capable of running spiking neural networks and achieving cognitive behaviors. By combining the power of AI algorithms and high-performance hardware with brain-inspired architectures, organizations can revolutionize industries and transform the way we live, work, and connect in the context of artificial intelligence. Furthermore, the brain's efficiency, characterized by sparse coding, analog computation, and spike-based communication, can be harnessed in neuromorphic systems, allowing for improved resource management, realistic interactions, and seamless user experiences in digital twin applications and the metaverse. Moreover, the capacity of associative memory, a crucial cognitive function of the brain, enables individuals to establish connections and draw inferences, facilitating the development of intelligent systems within the digital twin and metaverse realms [1–3].

Within the metaverse, innovations such as the octonion-based nonlinear echo state network elevate speech emotion recognition and user experiences [4]. Beyond emotion recognition, the metaverse's transformative impact extends to medical diagnosis. Employing deep learning algorithms for digital twinning of dental issues creates virtual facilities and medical services, fortified by blockchain integration [5]. These examples showcase how neuromorphic engineering and machine learning fuel the metaverse's evolution, shaping interconnected industries and elevating user experiences.

Recently, the implementation of the metaverse on clouds has encountered challenges related to long latency, security concerns, and centralized infrastructures. To address these issues, designing scalable metaverse platforms on the edge layer has emerged as a practical solution. However, the realization of edge-powered metaverse ecosystems depends heavily on high-performance intelligent edge devices. Neuromorphic engineering, which employs brain-inspired cognitive architectures to implement neuromorphic chips and tiny machine learning (TinyML) technologies, holds promise in enhancing edge devices for such emerging ecosystems. In this context, a super-efficient TinyML processor specifically designed for use in edge-enabled metaverse platforms has been developed and evaluated [6]. The processor incorporates a winner-take-all (WTA) circuit, implemented through a simplified leaky integrate-and-fire (LIF) neuron on an FPGA. The WTA architecture draws inspiration from the mini-column structure in the human brain, showcasing the potential of neuromorphic principles in edge devices. By employing the simplified LIF neuron, the resource consumption of the WTA architecture is significantly reduced, making it highly suitable for the proposed edge devices.

Neuromorphic engineering is a rapidly growing field that combines multiple disciplines to replicate the complex structure and organization of the brain in artificial systems. It utilizes insights from neurobiology to create integrated circuits that mimic the function and structure of biological nervous systems [7]. By imitating the brain's functional and structural features, neuromorphic engineering opens up exciting possibilities for developing novel computing architectures capable of running spiking neural networks and achieving cognitive behaviors [8]. The brain maintains its efficiency through various methods, including sparse coding, analog computation, and a spike-based communication system. Additionally, its ability to perform tasks in parallel and its adaptability contribute to its effectiveness in overcoming noise-related issues. In this regard, neuromorphic systems facilitate the accessibility to brain-inspired architectures on a chip [9]. Associative memory, a crucial cognitive function of the brain, involves the capacity to learn and recall the connections between different elements. It allows individuals to unconsciously establish associations and draw inferences from diverse experiences or occurrences.

Several neural associative memory models and structures have been developed, including the Hopfield neural network (HNN) [10] and the bidirectional associative memory (BAM) [11]. These architectures serve as fundamental associative memory systems and utilize binary neurons in their implementation. Nonetheless, the existing architectures mentioned above face limitations in terms of storage capacity. To address this, the clique-based neural network (CBNN) [12] has emerged as an alternative solution, capable of efficiently storing a significant amount of binary data using binary synapses and neurons. Moreover, recently introduced spiking associative memory architectures have demonstrated substantial advancements in terms of both storage capacity and reliable retrieval of stored information, outperforming HNN, BAM, and CBNN in these aspects [10–15].

Spiking neurons are fundamental units of computation in the brain that enable the transmission and processing of information through electrical impulses called spikes [16]. These spiking neurons generate action potentials, or spikes, in response to specific stimuli or inputs. These spikes are characteristic of neurons and play a crucial role in how the brain functions. The firing dynamics of spiking neurons have been the subject of extensive research, and various computational models have been developed to explain their behavior. One of the best-known computational models for spiking neurons is the Hodgkin–Huxley model. The Hodgkin–Huxley model is a mathematical representation that describes the

firing dynamics of spiking neurons. It takes into account various biophysical properties of neurons, such as the voltage-dependent conductance of ion channels [17]. The Hodgkin–Huxley model is widely used because it accurately captures the complex dynamics of spiking neurons and provides insights into how different factors, such as ion channel conductance, influence the generation and propagation of action potentials [18].

The LIF model is another important computational model used to describe the behavior of spiking neurons. It is a simpler model compared to the Hodgkin–Huxley model but still captures essential characteristics of spiking neurons. The LIF neuron model assumes that the membrane potential of a neuron integrates inputs from other neurons and external stimuli over time [19]. Once the membrane potential reaches a certain threshold, the neuron generates an action potential or spike. The LIF neuron model considers the leakage of current across the neuronal membrane, which causes the membrane potential to decay over time. This model is popular in computational neuroscience due to its simplicity and computational effectiveness.

In this paper, a new neuromorphic architecture called TOM is presented. The proposed architecture is compared to a conventional floating-point architecture to facilitate a comparative analysis. To implement the WTA modules, a simplified LIF neuron model is utilized as the core component [14,15]. To classify input patterns, the system employs multiple spiking WTA neural networks, where N is greater than 1. Each individual WTA module identifies the most stimulated neuron as the winner, subsequently eliciting spikes. The WTA's inherent feature facilitates the incorporation of sparse coding methodologies, wherein only a limited subset of neurons is collectively activated within the COM system [13]. The utilization of sparse coding in the COM architecture makes it well-suited for storing a vast amount of data. Furthermore, the interconnection of WTA modules through lateral excitatory synapses enhances the robust retrieval of stored information. The unique characteristics of the COM architecture position it as an ideal candidate for integration into a neuromorphic system, allowing it to leverage its distinctive attributes for effective implementation. The essential aspects of our study can be summarized in the following manner:

- **TOM Framework**: Innovative architecture inspired by COM for efficient digital device implementation.
- **Innovative Neural Architecture**: Introduction of DLBS architecture enhancing TOM efficiency.
- **Advanced Recognition Mechanism**: XNOR gate utilization for improved pattern recognition.
- **Detailed Test and Evaluation**: Comprehensive testing under varying conditions, including high noise and message erasure.
- **Integration of Synaptic Plasticity Rules**: Incorporation of STDP and Hebbian rules for enhanced cognitive capabilities.

2. Background and Motivation

The proposed features of the COM architecture have been previously discussed in [13]. Nonetheless, this paper provides a comprehensive review of the principles underlying the implementation of the COM architecture, including the concepts of message memory and message retrieval. Columnar-organized memory is a theoretical framework that proposes an organizational structure for neural networks or brain regions involved in memory processing. This organizational structure is inspired by the columnar organization observed in certain regions of the brain, such as the neocortex [20]. In the columnar-organized memory framework, neural networks or brain regions are organized into columns, which are vertical arrangements of neurons with similar functional properties. These columns are believed to play a crucial role in the storage and retrieval of information, as well as the formation of associations between different elements of memory. Figure 1 illustrates the structure of COM, consisting of multiple spiking WTA modules interconnected by lateral excitatory synapses. A WTA module comprises three distinct components: an input layer, an output layer, and lateral inhibitory connections that interconnect the output neurons via a reset sig-

nal. The synaptic connections create a neural link connecting the neurons of the input layer to those in the output layer of the WTA module. The output layer comprises a collection of LIF neurons that are interconnected through inhibitory connections. Furthermore, the separate WTA modules are interconnected by means of lateral excitatory synapses. During the message memory process, modifications occur in the weight values of the synaptic connections that transmit signals from the input to the output layers of a WTA module. Additionally, adjustments are made to the lateral excitatory synapses.

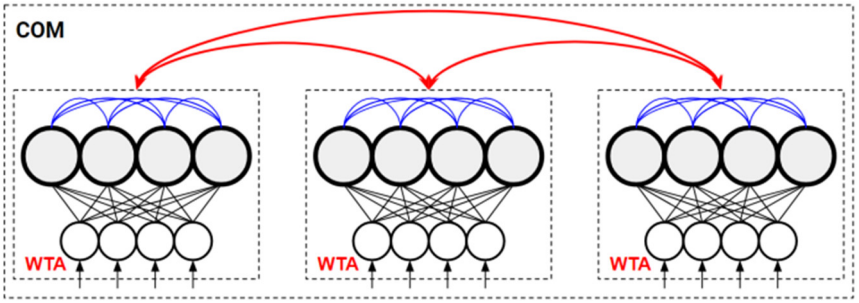

Figure 1. The architecture of the COM architecture [13] comprising the WTA modules. The blue lines show the inhibitory connections, and the red lines indicate the excitatory connections between WTA modules.

Sections 2.1 and 2.2 delve into detailed discussions on message memory and message retrieval, respectively, providing comprehensive insights into these aspects.

2.1. Message Memory

The message M comprises a 2D array consisting of N pattern vectors (M = {$p1$, $p2$, $p3$, . . ., pN.}). Each pattern p is a binary vector of length l, where the elements range from 0 to 1 (p = [$x1$, $x2$, $x3$, . . ., xl]). The number of pattern vectors N in the message matrix corresponds to the number of WTA modules in the COM architecture. (Pattern p is the specific vector associated with the $\boldsymbol{i_{th}}$ WTA module, denoted as $\boldsymbol{WTA_i}$.) Additionally, the length of the pattern l is established by considering the quantity of input neurons in the WTA's input layer (refer to Figure 1). The message memory process in the COM architecture comprises two key steps, pattern storage and pattern association. During the pattern storage step, each WTA module is trained using a pattern set consisting of N patterns (N corresponds to the number of WTA modules). In Figure 2, three pattern sets ($P1$, $P2$, $P3$) are utilized to train their respective WTA modules ($\boldsymbol{WTA_1}$,$\boldsymbol{WTA_2}$, $\boldsymbol{WTA_3}$). Each output neuron $\boldsymbol{n_k}$ of a WTA module is trained with its corresponding pattern $\boldsymbol{p_k}$ from the pattern set P. Concerning this, an STDP training algorithm is used to train the synaptic weights matrix of a WTA neural network. The STDP rule is a training algorithm that adjusts synaptic weights W based on the given patterns p in a spiking WTA [21]. The weight adjustment Δw is determined by the disparity between the spike times of the presynaptic and postsynaptic neurons [22]. The core equation of the STDP rule can be expressed as follows:

$$\Delta w = \begin{cases} \theta^{+} e^{-\Delta t/\tau^{+}} if\ \Delta t > 0 \\ -\theta^{-} e^{\Delta t/\tau^{-}} if\ \Delta t < 0, \end{cases} \quad (1)$$

where Δt represents the temporal gap between the occurrence of presynaptic and postsynaptic spikes ($\Delta t = \boldsymbol{t_{post}} - \boldsymbol{t_{pre}}$). The parameters of $\boldsymbol{\theta^{+}}$ and $\boldsymbol{\theta^{-}}$ correspond to the upper and lower limits of Δw, while $\boldsymbol{\tau^{+}}$ and $\boldsymbol{\tau^{-}}$ represent constants values. After the training procedure, the final synaptic weights matrix of the WTA is analogous to the input patterns. During the pattern association step, the COM structure stores each M message by establishing connections through the intraneuronal excitatory connections. In order to preserve a message matrix M = {$p1$, $p2$, $p3$, . . ., pN}, the output neuron associated with a learned pattern in a WTA module (denoted as $p \in M$) is connected to another output

neuron of a distinct WTA module representing a different pattern (denoted as $p_k \neq i \in M$) via the lateral excitatory synapses [6]. The COM structure stores two hypothetical messages ($m1$ and $m2$), as depicted in Figure 2. In this scenario, the neurons representing the patterns of message $m1$ are interconnected through lateral excitatory synapses, thereby establishing associations among them. Additionally, message $m2$ has been appropriately stored, resulting in the formation of a clique. A clique refers to a group of N neurons, where each neuron corresponds to a specific trained WTA. Figure 2 depicts two groups of neurons represented as cliques which consist of $c_1 = \{A, b, \delta\}$ and $c_2 = \{C, a, \beta\}$.

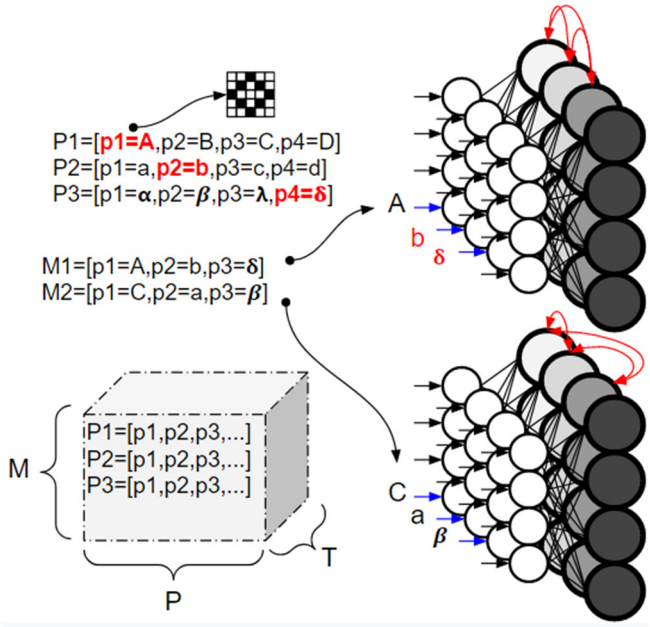

Figure 2. The storage and retrieval process of messages in a TOM architecture with three incorporated WTA networks. First, three sets of patterns are employed to train the corresponding WTA modules. The TOM structure stores two messages, M1 and M2, by adapting the lateral excitatory connections. The retrieval mechanism of the message M1 is achieved by applying a designated pattern, referred to as m', to the TOM.

2.2. Message Retrieval Process

During the process of message retrieval, the reactivation of a specific group is triggered to recall a stored message m_i. Essentially, the output neurons of WTA modules within a clique possess inherent memory of recently learned patterns and their associations [21]. Each winning neuron represents a stored message m that closely resembles the input message m'.

Figure 2 shows how COM recovers a hypothetical partially erased message m' through the message retrieval process and produces spikes. Subsequently, the winning neuron A transmits these spike signals through the excitatory connections to other neurons within the clique $c = \{A, b, \delta\}$. This indicates that the activation of the neuron A leads to the activation of the corresponding neurons b and δ. The set of activated neurons (A, b, and δ) shows our original message ($m1$) before it was erased, which closely resembles the message m'(Figure 3).

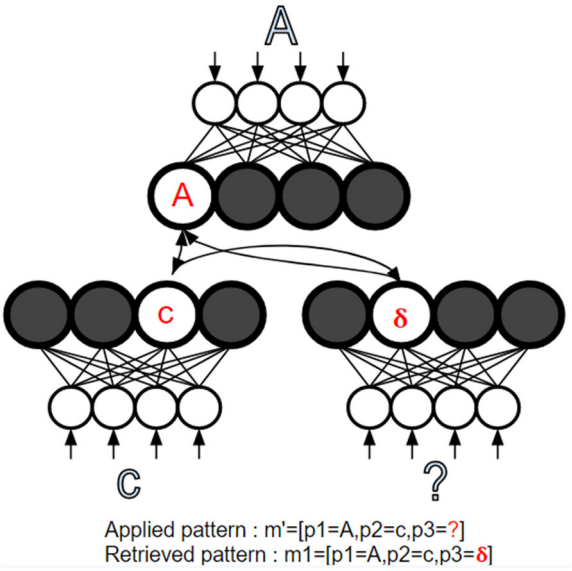

Figure 3. Illustration of the process of message storage and retrieval in a TOM architecture incorporating three WTA modules. Specifically, it demonstrates the retrieval of message m1 by applying a designated pattern $\boldsymbol{m'}$ to the system.

3. Implementation Method

In this section, the hardware architecture of TOM is presented, which consists of three design levels. In Level I, we introduce an optimized LIF neuron model and compare it with a floating-point neuron architecture. In Level II, we propose and implement a WTA architecture. Lastly, in the final level, we develop the TOM hardware architecture, utilizing the WTA modules and excitatory synaptic connections.

3.1. Level I: LIF Neuron Hardware Architecture

At the first level, a neuron architecture is designed which has been optimized for implementing on FPGA. The Euler method has been used in order to simplify Equation (1) [14]:

$$u[n] = \left(\frac{\tau_m}{\tau_m + 1}\right) u[n-1] + \frac{R}{\tau_m + 1} \times I[n] \tag{2}$$

Let us assume $\left(\frac{\tau_m}{\tau_m+1}\right) = \alpha$, and $\frac{R}{\tau_m+1} = \beta$. Equation (2) can be written as follows:

$$\boldsymbol{u[n] = \alpha u[n-1] + \beta I[n], 0 < |\alpha| \leq 1} \tag{3}$$

where α depends on $\boldsymbol{\tau}$. The term β determines the amplitude of the neuron input. In frequency-domain space, Equation (3) is shown as follows:

$$\begin{gathered} \boldsymbol{u[z] = \alpha z^{-1} u[z] + \beta I[z]} \\ \boldsymbol{if\, I[n] = \delta[n]\ then\ H[z] = \frac{\beta}{1 - \alpha z^{-1}} z} \end{gathered} \tag{4}$$

where $H[z]$ indicates the impulse response function of the LIF neuron. By using inverse Z-Transform, $h[n]$ is defined as

$$H[z] = \frac{\beta}{1 - \alpha z^{-1}} \overset{z^{-1}}{\leftrightarrow} h[n] = \beta \alpha^n u[n] \tag{5}$$

Let us assume the step function as the input of neurons which is derived by corresponding pixels in the input pattern (more information discussed at Section 4):

$$\begin{gathered} \boldsymbol{H[z] \cdot u[z] = \frac{\beta}{1-\alpha z^{-1}} \times \frac{1}{1-z^{-1}}} \\ \boldsymbol{s[n] = \beta \frac{1-\alpha^{n+1}}{1-\alpha} \cdot u[n]} \end{gathered} \tag{6}$$

Consequently, the condition for choosing the threshold voltage in respect of maximum neuron membrane potential is

$$y(+\infty) = \frac{\beta}{1-\alpha}\ ,\ n > 0 \tag{7}$$

By analyzing Equation (7), the maximum required bits to show membrane potential can be determined. This equation consists of two parts, which are $A = \frac{\beta}{1-\alpha}$ and $B = \alpha^{n+1}$. If it is possible to rewrite terms A and B as a multiple of 2, then:

$$\boldsymbol{A = 2^m, B = 2^{-a(n+1)}, m, a \in \{Z\} > 0} \tag{8}$$

Now Equation (8) can be rewritten as follows:

$$s[n] = 2^m, B = 2^{-\alpha(n+1)} \tag{9}$$

Since m and a are positive integer numbers, and Equation (3) is a monotonically increasing function, to represent the difference between these two terms one N-bit register is needed, in which N is $\boldsymbol{\alpha(n_{max}+1)}$. $\boldsymbol{n_{max}}$ is the time when a spike occurs that is determined as follows:

$$\boldsymbol{s[n] = \beta \frac{1-\alpha^{n+1}}{1-\alpha} \geq \gamma, n > 0} \tag{10}$$

$$\boldsymbol{\log_{\alpha}\left[1-\gamma\frac{1-\alpha}{\beta}\right] - 1 \geq n_{max}} \tag{11}$$

where γ is the neuron spiking threshold. Figure 4 shows the neuron step response, which is implemented using a single shift register. Figure 5 represents the architecture of a single neuron unit. The neuron hardware utilizes only two flip-flops and one LUT that is extremely optimized and appropriate to implement on FPGAs.

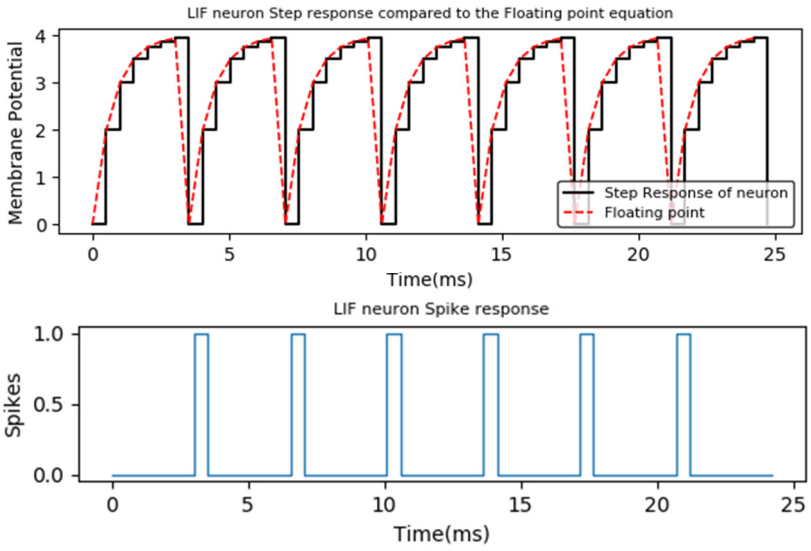

Figure 4. Neuron step response and its corresponding spikes when the membrane potential reaches the threshold value; $\alpha = 0.5$, $\beta = 2$.

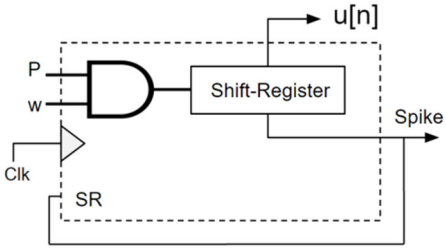

Figure 5. A single neuron hardware architecture. P and W represent the input pattern and corresponding weight matrix, respectively.

3.2. Level II: WTA Hardware Architecture

At the second level, a WTA module architecture is implemented. The introduced LIF neuron at level 1 is dedicated to developing the first layer of the WTA. Additionally, a new floating-point LIF neuron is represented to utilize in the second layer in order to increase the accuracy of the WTA module. In order to implement the floating-point neuron, two multipliers, one accumulator, one register, and one comparator unit are needed. By dividing both sides of Equation (3) the number of multiplies can potentially be reduced to one:

$$\frac{u[n]}{\beta} = \frac{\alpha}{\beta}u[n-1] + I[n] \tag{12}$$

By defining $\frac{u[n]}{\beta}$ as $u_{new}[n]$ and $\frac{\alpha}{\beta} = \eta$, the new form of Equation (12) leads to the following result:

$$u_{new}[n] = \eta \cdot u[n-1] + I[n],\ 0 < |\eta| \leq 1 \tag{13}$$

By choosing $\gamma_{new} = \frac{\gamma}{\beta}$ and considering Equation (11), Equation (13) has the same behavior and response as Equation (3), but the implementation of this new scaled equation only needs one multiplier. This technique has reduced the number of the required multipliers by a factor of two. The hardware architecture for the floating-point LIF neuron is represented in Figure 6.

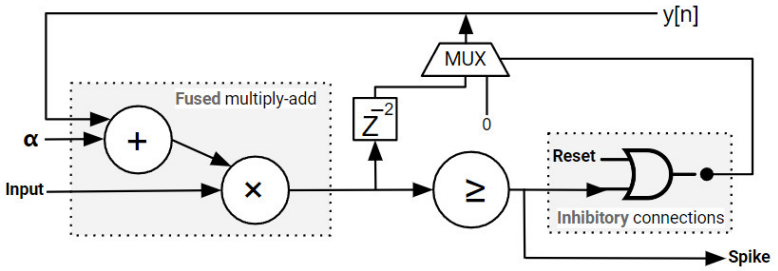

Figure 6. Hardware architecture of the floating-point LIF neuron. The inhibitory connection block shows how the winner neuron resets other neurons at the output layer. The z^{-2} term compensates latency because of the comparator unit.

The WTA consists of two layers. The first layer utilizes 25 neurons to receive inputs. The second layer performs as a classifier. This module is trained through the pattern storage process. The outputs of neurons for the second layer are connected to each other through inhibitory synaptic connections. The winner neuron resets other neurons' membrane potential using a reset connection. The STDP learning algorithm is used to train WTA. It performs in software by using the off-chip method. The synaptic weights matrix between these two layers is fed to the network manually. Figure 7 shows the structure of a WTA module.

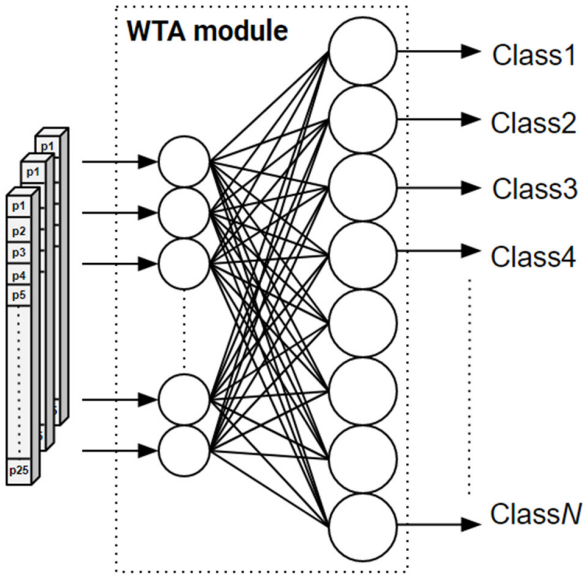

Figure 7. The architecture of the proposed WTA neural network. Each class is trained through the pattern storage procedure, which has been discussed in Section 2.

3.3. Level III: TOM Hardware Architecture

The hardware implementation of the LIF neuron and WTA module has been introduced at the first and second levels of design. Our methodology involves the utilization of the ex situ method to store a single message in the TOM structure. By implementing the STDP algorithm in software, we calculate the feedforward synaptic weight vector that establishes connections between the two layers of the WTA module. The retrieval of an erased message in the COM structure is primarily facilitated by the excitatory connections. To serve as an excitatory connection in TOM, we propose a multi-input OR gate. The architecture of TOM is illustrated in Figure 8.

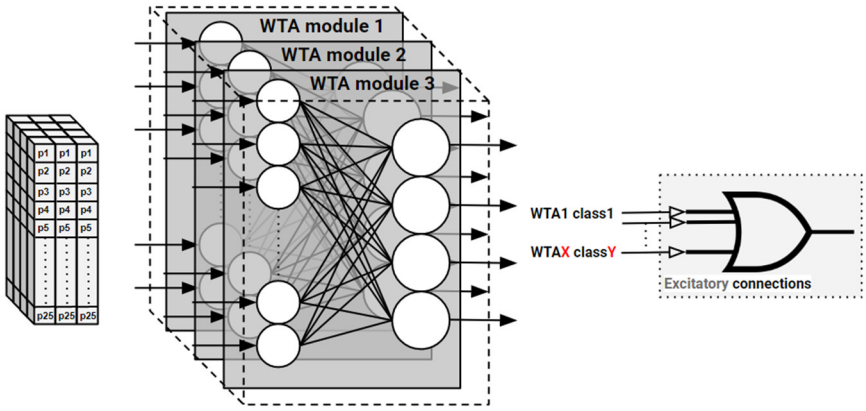

Figure 8. The proposed hardware architecture of the COM. The excitatory connection block shows how COM operates to retrieve an erased message.

Each message consists of a combination of patterns. In order to retrieve a correct set of patterns which relate together to form a message, these patterns should connect together through the message retrieval process. Therefore, an OR gate connects each set of patterns together to activate each other through this route. For example, let us consider the $m1$ message which is shown in Figure 3. If an erased message like m' fed to the network, then, to retrieve this message an OR gate should be organized to connect each class of message $m1$ together. Consequently, each pattern of the erased message can restore the rest of its associated patterns only by using an OR gate.

3.4. A Novel COM Architecture Based on Digital Logic

In this section, a novel architecture which is named digital-logic-based system (DLBS) is introduced. This is an effort to implement an efficient TOM architecture by using previous experiences. The throughput and latency of the proposed hardware structure are extremely improved. Let us take a closer look at the defining equation of the LIF neuron model:

$$spike_1[n] = (\mathrm{p}_i \cdot \mathrm{w}_i) \cdot \sum_{k=0}^{+\infty} \delta[n - kN_1] \tag{14}$$

where $spike_1[n]$ shows the spike times of the input layer, and $N_1 = n_{max} + 1$, respectively. $(\mathrm{p}_i \cdot \mathrm{w}_i)$ shows the AND operation of input with the first synaptic weight matrix. Considering $x[n]_2$ as the output of the first layer and input of the second layer interchangeably, we obtain:

$$x_2[n] = \sum_{k=0}^{+\infty} \sum_{i=0}^{n_1-1} (p_{1,i} \cdot W_{1,i} \cdot W_{2,i}) \cdot \delta[n - KN_1] \tag{15}$$

where $p_{1,i}$ is the input of the first layer and $W_{1,i}$, $W_{2,i}$ determine the first and second layers' corresponding synaptic weights. n_1 shows the number of neurons at the input layer. The intensity and amplitude of the total white pixels in our pattern are determined by $S_{1,j}$ as follows:

$$S_{1,j} = \sum_{i=0}^{n_1-1} (p_{1,i} \cdot W_{1,i} \cdot W_{2,i}) \tag{16}$$

By convolving the impulse response of the LIF neuron model Equation (5) with Equation (15), we obtain:

$$y_2[n] = \beta_2 \alpha_2^n u * S_{1,j} \sum_{k=0}^{+\infty} \delta[n - KN_1] \tag{17}$$

$$y_2[n] = \beta_2 S_{1,j} \frac{\alpha_2^n - \alpha_2^{-N_1}}{1 - \alpha_2^{-N_1}} \geq \gamma,\ 0 < \alpha_2 < 1 \tag{18}$$

Once again, γ determines the threshold value of the neurons. Utilizing Equation (18) to determine n_{max} leads to the following result:

$$n_{max} = N_2 - 1 \leq \log_{\alpha_2} \left[\frac{\gamma_2 \left(1 - \alpha_2^{N_1}\right)}{\beta_2 S_2} + \alpha_2^{-N_1} \right] \tag{19}$$

The obtained Equation (19) illustrates that the output neuron of the second layer is an impulse train with n_{max} period. The derivative of this equation with regards to the S_2 is:

$$\frac{\partial N_2}{\partial S_2} = \frac{-\beta_2 \gamma_2 \left(1 - \alpha_2^{-N_1}\right)}{\beta_2 \gamma_2 S_2 \left(1 - \alpha_2^{-N_1}\right) + \alpha_2^{-N_1} \beta_2 S_2} \tag{20}$$

Equation (20) shows that the spike frequency of the output neurons directly relates to the amplitude of the S term. Consequently, S_2 can be used as a hyperparameter to recognize the winner neuron in WTA. In the previous architecture, the AND gate is utilized to recognize white pixels in patterns and ignore black pixels. This means that the black pixels do not participate in the process of recognizing a particular pattern. In this architecture, the input patterns and synaptic weights of the first layer are passed through an XNOR gate. XNOR gates have the ability to distinguish the white pixels from black pixels in the patterns. Therefore, to recognize a pattern, not only is the location of its white pixels essential, but also black pixels are not ignored and take part in the recognition process. This method enhances network accuracy by a factor of two. The high-level diagram of this new architecture is shown in Figure 9.

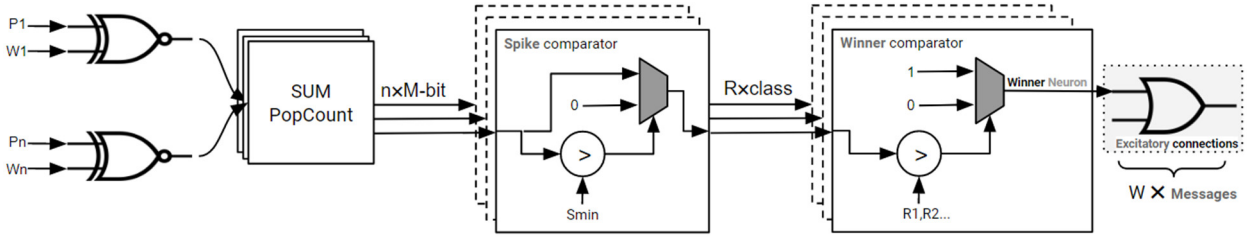

Figure 9. The high-level diagram of the novel architecture.

4. Test Procedure

To evaluate the effectiveness of the proposed columnar-organized memory architecture, a test procedure can be implemented. It is essential to note that our dataset has been generated using Python code, and Gaussian noise has been applied to it using our custom script developed in Python. Furthermore, the message retrieval process is also conducted in Python to validate the accuracy, F1-score, and loss of our proposed models. The test procedure involves several steps. First, a dataset is prepared that includes a variety of inputs and corresponding desired outputs. The inputs can represent different types of information or stimuli, and the desired outputs can be the expected responses or actions associated with those inputs. The prepared dataset is then used to train the TOM architecture. This training process involves adjusting the synaptic weights and parameters of the spiking neurons in order to optimize the performance of the architecture for storing and retrieving information.

Once the training process is complete, the effectiveness of the architecture can be evaluated through testing. During the testing phase, various inputs are presented to the columnar-organized memory architecture and the corresponding outputs or responses generated by the architecture are compared to the desired outputs. This allows for an assessment of the accuracy and reliability of the memory storage and retrieval processes within the columnar-organized memory architecture. Our dataset consists of 25 English alphabets and it is used to train our WTA modules. Each module consists of a combination of four alphabets to test TOM architecture. Our test procedure is divided into two parts. First, one of the combinations is missing and we use the TOM architecture to retrieve the original message. In the second part of the test, each message has two random elements lost in their architecture. Each test procedure iterates for 100 epochs in each module (Figure 10).

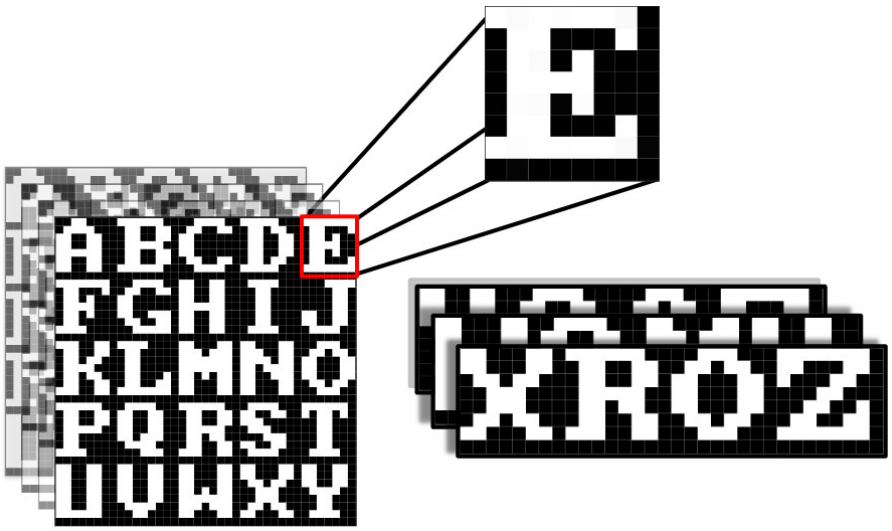

Figure 10. The proposed dataset of 25 alphabets which consist of "8 × 8"images. Each image for our test messages consists of four alphabets for the validation purpose.

The process of adding noise is as follows: a Gaussian grayscale noise has been added to the dataset with a percentage between 0% and 50%. Each message in two separated parts has some lost element as shown in Figure 11. The loss effect can show the power of reconstruction of a TOM architecture inspired by COM.

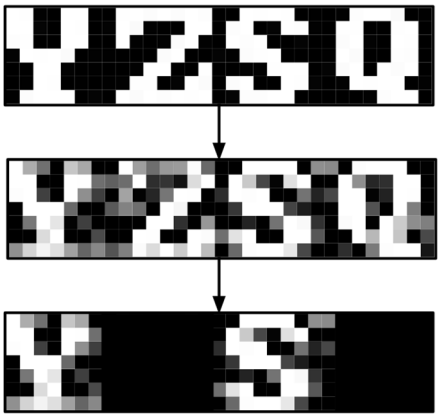

Figure 11. This image shows the procedure of added Gaussian noise and the message loss effect which is transmitted into the TOM module as the input.

5. Results and Discussion

In this section, we present the simulation outcomes and engage in a comprehensive discussion regarding them. The TOM architecture incorporates the principles of STDP, a fundamental mechanism of synaptic plasticity in neural systems, as well as Hebbian rules. The STDP influences synaptic strength based on the precise timing of neuronal spikes. When a presynaptic neuron fires before a postsynaptic neuron, the synaptic connection between them is potentiated, while firing shortly after weakens the connection. Hebbian rules reinforce synaptic connections when the presynaptic and postsynaptic neurons are active simultaneously, promoting the formation of functional circuits.

By incorporating STDP and Hebbian rules into the training process of the TOM architecture, the system can adapt and refine its synaptic connections based on the precise timing and correlation of neuronal activity. This enables the TOM architecture to learn and encode information efficiently, facilitating the categorization and processing of input patterns.

These rules are used to calculate the synaptic weights, which determine the strength or efficacy of synapses. After the weights are calculated, they are loaded into the TOM registers. In this case, the synaptic weights determined by STDP and Hebbian rules are calculated and saved into the weight registers, allowing the TOM architecture to learn and adapt accordingly. As a result, the TOM architecture is prepared to effectively handle and retrieve messages.

The simulation results validated the successful implementation and proficient operation of the proposed TOM architecture throughout all design stages, including the proposed neuron, WTA module, and the overall TOM system. When subjected to message retrieval tasks using noise levels ranging from 0% to 30% and partially erased messages, the average message retrieval rates for the simulations were approximately 0.9 and 0.8, respectively. This result demonstrates the robustness of the TOM architecture and its ability to perform adequately under less-than-ideal conditions. However, the retrieval process of noisy messages with more than 30% noise, as well as partially erased messages, did show some degradation in performance (as shown in Figure 12).

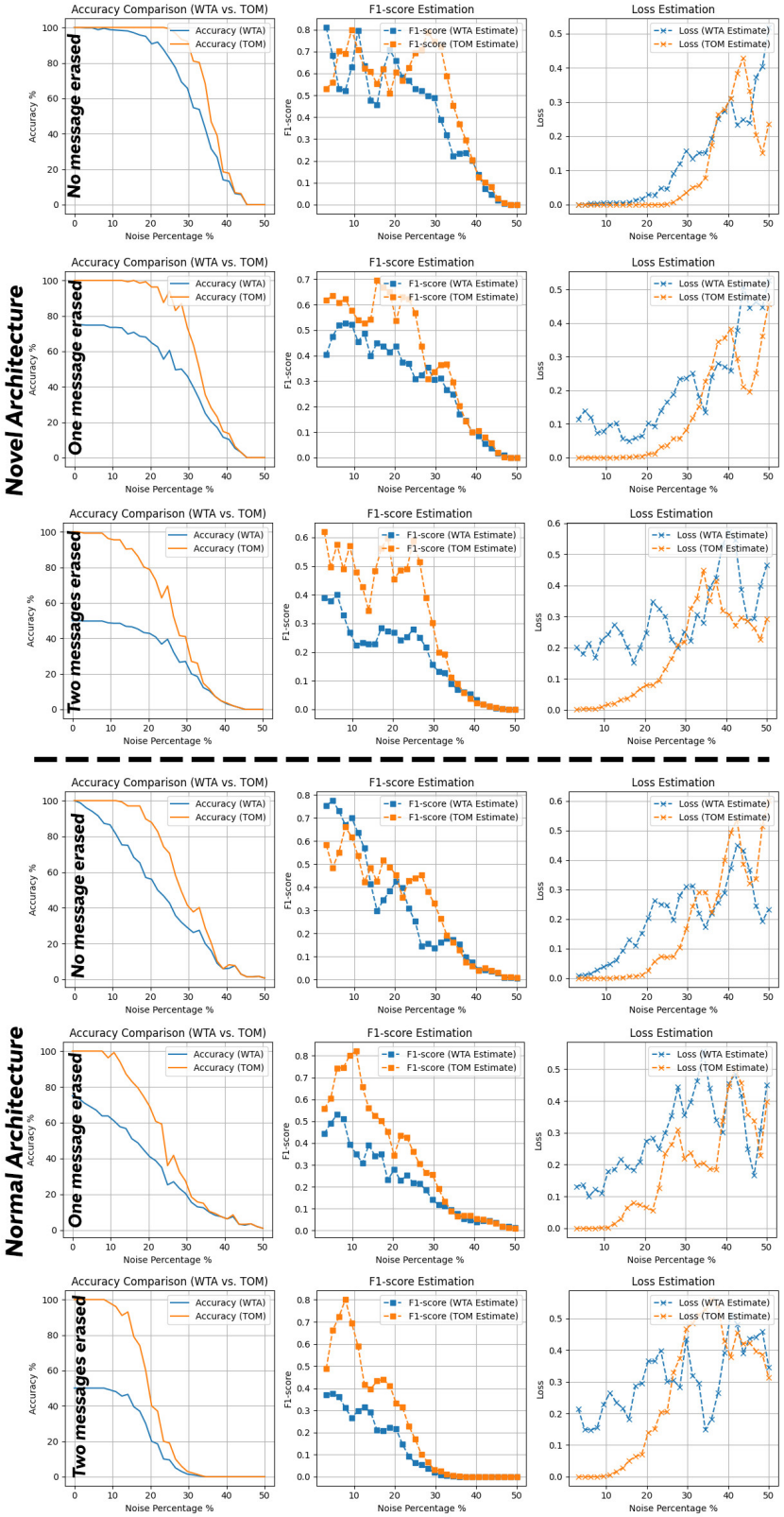

Figure 12. The result of the comparison between WTA and TOM architecture in terms of message loss and retrieval of the original message. Each test is divided into two parts. One set of results concerns the "Novel" architecture and the other one the "Normal" architecture. The F1-score and loss have been estimated using our observed value for the accuracy part in different tests. (In order to have smoother plots, a moving average filter has been applied to the F1-score and loss estimation plot).

Our experiments in three different test settings with different levels of noise on the TOM show that both the "Novel" and "Normal" designs perform better than the WTA module. The F1-score helps us evaluate the balance between precision and recall in classification tasks, while the loss metric quantifies the disparity between predicted outputs and actual target values, giving us valuable insights into overall predictive accuracy. These observations are especially significant in high-noise scenarios, where our architecture shows a substantial improvement. Moreover, our message retrieval capability remains robust even when a portion of the pattern is lost.

The suggested designs for the TOM neural network encompass both conventional and innovative neural network architectures, and they are entirely novel. There is no prior record of these approaches being put into practice within digital systems. Consequently, there is no existing implementation that qualifies for a comparison with our own, either in terms of functionality or performance. This implementation paves the way for a new era in the development of neurological systems on digital platforms, such as FPGA. This advancement holds significant potential for the implementation of neuromorphic and spiking neural network architectures, drawing inspiration from neuroscience and the human brain.

As mentioned earlier, both the "Novel" and "Normal" architectures outperform the WTA module. To compare these two architectures, we've included Figure 13. As can be seen in the plots, the "Novel" architecture continues to outperform, enhancing our TOM implementation concept further. When we compare them using metrics like the F1-score and loss, the "Novel" architecture consistently performs better, and it also has the potential to use fewer digital resources in its implementation, For instance, by the results obtained from a single test case where no messages of the pattern have been erased it can inferred that the Novel architecture perform drastically better in different scenarios compare to the Normal architecture. The introduction of this Novel architecture improves the represented TOM architecture and make it applicable to utilize in digital systems (Figure 13).

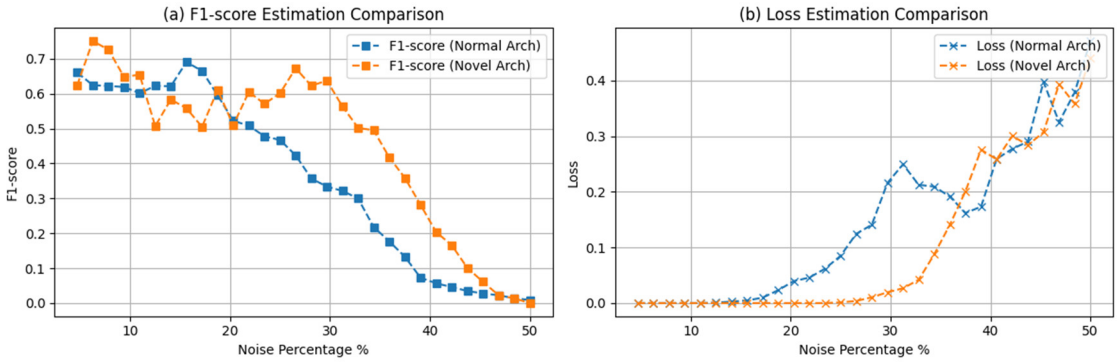

Figure 13. (**a**) illustrates the F1-score estimation comparison of the "Normal" and "Novel" architectures on the chart. (**b**) shows the loss estimation comparison of these two architectures in the same scenario which is "No message erased". (Please consider that, in this analysis, we have used the moving average technique on the result obtained as accuracy to estimate the f1-score and loss).

6. Conclusions

This research provides a promising glimpse into the future of neuromorphic engineering. We have delved into the potential of the TOM architecture, observing how remarkably it integrates neural and synaptic principles. Our simulation results are compelling, showing the TOM framework's resilience under less-than-optimal conditions and demonstrating its ability to adapt and optimize message retrieval, even in noisy environments. Although performance degradation was observed under noise levels exceeding 30% or with partially erased messages, we anticipate that further refinement of the TOM architecture may mitigate these limitations. This study has shed light on the critical role that STDP and Hebbian learning principles, and also our proposed retrieval architecture play in neuromorphic

computing, guiding the way for future advancements in this field. As we strive to replicate brain-like functionalities in artificial systems, the application of the TOM architecture in neuromorphic chips for intelligent edge devices holds significant promise. There are exciting implications for the ever-evolving metaverse, suggesting new opportunities for high-performance computation and authentic user experiences that this architecture could foster in the near future. Our research represents a pioneering step toward advancing neurological systems on digital platforms, such as FPGA, with potential applications in neuromorphic and spiking neural network architectures inspired by neuroscience and the human brain. Further exploration and research into the TOM architecture will continue to shape the future of neuromorphic engineering.

Author Contributions: Conceptualization, A.K.N. and M.J.; methodology, A.K.N. and M.J.; software, A.K.N.; validation, A.K.N.; formal analysis, A.K.N.; investigation, A.K.N. and M.J.; resources and writing—original draft preparation, A.K.N. and M.J.; writing—review and editing, M.J.; visualization, A.K.N.; supervision, S.B.S.; project administration, S.B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shrestha, A.; Fang, H.; Mei, Z.; Rider, D.P.; Wu, Q.; Qiu, Q. A Survey on neuromorphic computing: Models and hardware. *IEEE Circuits Syst. Mag.* **2022**, *22*, 6–35. [CrossRef]
2. CSchuman, C.D.; Kulkarni, S.R.; Parsa, M.; Mitchell, J.P.; Date, P.; Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nat. Comput. Sci.* **2022**, *2*, 10–19. [CrossRef]
3. Jiang, Y.; Yin, S.; Li, K.; Luo, H.; Kaynak, O. Industrial applications of digital twins. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2021**, *379*, 20200360. [CrossRef] [PubMed]
4. Daneshfar, F.; Jamshidi, M. An Octonion-Based Nonlinear Echo State Network for Speech Emotion Recognition in Metaverse. *Neural Netw.* **2023**, *163*, 108–121. [CrossRef] [PubMed]
5. Moztarzadeh, O.; Jamshidi, M.; Sargolzaei, S.; Keikhaee, F.; Jamshidi, A.; Shadroo, S.; Hauer, L. Metaverse and Medical Diagnosis: A Blockchain-Based Digital Twinning Approach Based on MobileNetV2 Algorithm for Cervical Vertebral Maturation. *Diagnostics* **2023**, *13*, 1485. [CrossRef] [PubMed]
6. Khajooei, A.; Jamshidi, M.; Shokouhi, S.B. A Super-Efficient TinyML Processor for the Edge Metaverse. *Information* **2023**, *14*, 235. [CrossRef]
7. Yang, J.; Wang, R.; Ren, Y.; Mao, J.; Wang, Z.; Zhou, Y.; Han, S. Neuromorphic Engineering: From Biological to Spike-Based Hardware Nervous Systems. *Adv. Mater.* **2020**, *32*, e2003610. [CrossRef]
8. Mead, C. How we created neuromorphic engineering. *Nat. Electron.* **2020**, *3*, 434–435. [CrossRef]
9. Parhi, K.K.; Unnikrishnan, N.K. Brain-Inspired Computing: Models and Architectures. *IEEE Open J. Circuits Syst.* **2020**, *1*, 185–204. [CrossRef]
10. Hopfield, J.J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA* **1982**, *79*, 2554–2558. [CrossRef]
11. Kosko, B. Adaptive bidirectional associative memories. *Appl. Opt.* **1987**, *26*, 4947–4960. [CrossRef] [PubMed]
12. Gripon, V.; Berrou, C. Sparse Neural Networks With Large Learning Diversity. *IEEE Trans. Neural Networks* **2011**, *22*, 1087–1096. [CrossRef] [PubMed]
13. Shamsi, J.; Mohammadi, K.; Shokouhi, S.B. A Hardware Architecture for Columnar-Organized Memory Based on CMOS Neuron and Memristor Crossbar Arrays. *IEEE Trans. Very Large Scale Integr. Syst.* **2018**, *26*, 2795–2805. [CrossRef]
14. Lu, S.; Xu, F. Linear leaky-integrate-and-fire neuron model based spiking neural networks and its mapping relationship to deep neural networks. *Front. Neurosci.* **2022**, *16*, 857513. [CrossRef] [PubMed]
15. Wang, Z.; Guo, L.; Adjouadi, M. A Generalized leaky integrate-and-fire neuron model with fast implementation method. *Int. J. Neural Syst.* **2014**, *24*, 1440004. [CrossRef] [PubMed]
16. Fortuna, L.; Buscarino, A. Spiking Neuron Mathematical Models: A Compact Overview. *Bioengineering* **2023**, *10*, 174. [CrossRef]
17. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **1952**, *117*, 500–544. [CrossRef]
18. Piccinini, G. Computational explanation in neuroscience. *Synthese* **2006**, *153*, 343–353. [CrossRef]
19. Izhikevich, E. Which Model to Use for Cortical Spiking Neurons? *IEEE Trans. Neural Networks* **2004**, *15*, 1063–1070. [CrossRef]
20. Mountcastle, V.B. The columnar organization of the neocortex. *Brain* **1997**, *120*, 701–722. [CrossRef]

21. Masquelier, T.; Guyonneau, R.; Thorpe, S.J. Competitive STDP-based spike pattern learning. *Neural Comput.* **2009**, *21*, 1259–1276. [CrossRef] [PubMed]
22. Bi, G.-Q.; Poo, M.-M. Synaptic Modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* **1998**, *18*, 10464–10472. [CrossRef] [PubMed]