Received March 22, 2020, accepted March 29, 2020, date of publication April 6, 2020, date of current version April 21, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2985839*

An Overview of Neuromorphic Computing for Artificial Intelligence Enabled Hardware-Based Hopfield Neural Network

ZHEQI YU1, (Student Member, IEEE), AMIR M. ABDULGHANI1,2, (Senior Member, IEEE), ADNAN ZAHID1, (Student Member, IEEE), HADI HEIDARI1, (Senior Member, IEEE), MUHAMMAD ALI IMRAN1, (Senior Member, IEEE), AND QAMMER H. ABBASI1, (Senior Member, IEEE)



* James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.
* Department of Electrical and Computer Engineering, Sultan Qaboos University, Muscat 123, Oman

Corresponding author: Qammer H. Abbasi (qammer.abbasi@glasgow.ac.uk)

The work of Zheqi Yu was supported by the Joint Industrial Scholarship Between the University of Glasgow and Transreport Ltd., London. The work of Adnan Zahid was supported by the Engineering and Physical Sciences Research Council (EPSRC), Doctoral Training Grant (DTG) under Grant EPSRC DTG EP/N509668/1 Eng.

 **ABSTRACT** Compared with von Neumann’s computer architecture, neuromorphic systems offer moreunique and novel solutions to the arti cial intelligence discipline. Inspired by biology, this novel system has implemented the theory of human brain modeling by connecting feigned neurons and synapses to reveal the new neuroscience concepts. Many researchers have vastly invested in neuro-inspired models, algorithms, learning approaches, operation systems for the exploration of the neuromorphic system and have implemented many corresponding applications. Recently, some researchers have demonstrated the capabilities of Hop eld algorithms in some large-scale notable hardware projects and seen signi cant progression. This paper presents a comprehensive review and focuses extensively on the Hop eld algorithm’s model and its potential advancement in new research applications. Towards the end, we conclude with a broad discussion and a viable plan for the latest application prospects to facilitate developers with a better understanding of the aforementioned model in accordance to build their own arti cial intelligence projects.



 **INDEX TERMS** Neuromorphic computing, neuro-inspired model, Hop eld algorithm, arti cial intelli-gence.



**I. INTRODUCTION**

Nowadays, neuromorphic computing has become a popular architecture of choice instead of von Neumann computing architecture for applications such as cognitive processing. Based on highly connected synthetic neurons and synapses to build biologically inspired methods, which to achieve theoretical neuroscienti c models and challenging machine learning techniques using. The von Neumann architecture is the computing standard predominantly for machines. How-ever, it has signi cant differences in organizational struc-ture, power requirements, and processing capabilities relative to the working model of the human brain [1]. Therefore,

The associate editor coordinating the review of this manuscript and approving it for publication was Mostafa Rahimi Azghadi .



neuromorphic calculations have emerged in recent years as an auxiliary architecture for the von Neumann system. Neu-romorphic calculations are applied to create a programming framework. The system can learn and create applications from these computations to simulate neuromorphic functions. These can be de ned as neuro-inspired models, algorithms and learning methods, hardware and equipment, support sys-tems and applications [2].

Neuromorphic architectures have several signi cant and special requirements, such as higher connection and par-allelism, low power consumption, memory collocation and processing [3]. Its strong ability to execute complex compu-tational speeds compared to traditional von Neumann archi-tectures, saving power and smaller size of the footprint. These features are the bottleneck of the von Neumann architecture,

|  |  |  |  |
| --- | --- | --- | --- |
| VOLUME 8, 2020 | This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/ | 67085 |  |
|  |  |

Z. Yu *et al.*: Overview of Neuromorphic Computing

so the neuromorphic architecture will be considered as an appropriate choice for implementing machine learning algo-rithms [4].

There are ten main motivations for using neuromorphic architecture, including Real-time performance, Parallelism, von Neumann Bottleneck, Scalability, Low power, Foot-print, Fault Tolerance, Faster, Online Learning and Neu-roscience [1]. Among them, real-time performance is the main driving force of the neuromotor system. Through paral-lelism and hardware-accelerated computing, these devices are often able to perform neural network computing applications faster than von Neumann architectures [5]. In recent years, the more focused area for neuromorphic system develop-ment has been low power consumption [5] [7]. Biological neural networks are fundamentally asynchronous [8], and the brain’s ef cient data-driven can be based on event-based computational models [9]. However, managing the commu-nication of asynchronous and event-based task in large sys-tems is a challenging in the von Neumann architecture [10]. The hardware implementation of neuromorphic computing is favourable to the large-scale parallel computing architecture as it includes both processing memory and computation in the neuron nodes and achieves ultra-low power consump-tion in the data processing. Moreover, it is easy to obtain a large-scale neural network based on the scalability. Because of all aforementioned advantages, it is better to consider the neuromorphic architecture than von Neuman for hardware implementation [11].

The basic problem with neuromorphic calculations is how to structure the neural network model. The compo-sition of biological neurons is usually composed of cell bodies, axon, and dendrites. The neuron models imple-mented by each component of the speci ed model are divided into ve groups, based on the type of model being distinguished by biologically and computationally driven.

1. Biologically Plausible [12]: Speci cally simulate the behaviour type present in the biological nervous system. Such as a Hodgkin-Huxley model, the understanding of neuronal activity from the perspective of ions entering and leaving the neuron is based on a four-channel nonlinear differential equa-tion [13]. Other one is Morris Lecar model, which depends on a two-dimensional nonlinear equation for effective and simple calculation and implementation [14]. Meanwhile, a calcium-based model is a simpli ed biologically plau-sible implementation, providing the link concept between stimulation protocols, calcium transients, protein denoting cascades and induced synaptic changes [15]. Finally, for a Galves-Löcherbach model, it combines the spiking levels with biological rationality, and a model with inherent ran-domness [16].
2. Biologically-Inspired [17]: Ignore biological rational-ity to replicate biological nervous system behavior. Such as the Izhikevich model, has both simplicity and the ability to replicate biologically precise behavior [18]. Other one is Hindmarsh-Rose model, which satisfactorily explains the

67086

dynamics of pulse ring, cluster ring, and chaotic behavior of neurons [19].

1. Neuron and other Biologically- Inspired Mecha-nism [20]: Neuron models include other biologically inspired components.
2. Integrate and-Fire Neurons [21]: Biology-inspired spike neuron model.
3. McCulloch-Pitts Neurons [22]: The excitation function of neurons is a strict threshold function on the neuron models.

Similarly, synaptic models can be divided into two cat-egories. One of the synaptic models is bio-inspired synap-tic implementations that include spike-based systems and feed-forward neural networks [23]. For more complex synap-tic models, another common approach is based on plasticity mechanism that depends on the intensity or weight of a neuron to change over time [24].

Different network topologies are required for neuro-morphic systems. In network models, the more popular implementations are feed-forward neural networks, such as multilayer sensing, and other architectures include Recurrent neural networks, Stochastic neural networks [25], spiking neural networks [26], arti cial neural network cellular neu-ral networks [27] and pulse-coupled neural networks [28], cellular automata [29], fuzzy neural networks [30]. Hop-eld network as the RNN network architecture is especially common in the early implementation of neural morphology, which is consistent with the neural network research trend, there are more recently implementations now. Such as graph partition [31], fault detection [32] and data classi cation [33], etc.

For the algorithm, the learning method should match each requirement differences on speci c network topology, neuron model or other features of network model. In the algorithm learning process, supervised learning is gener-ally not considered as an online method. The widely used algorithm for programming neuromorphic systems is back-propagation technique. In contrast, unsupervised learning is based on self-organizing maps or self-organizing learning rules.

Neuromorphic implementation based on high-level stan-dards is to divide hardware implementation into three categories: digital, analog, and hybrid analog/digital plat-forms [17]. The analog system takes advantage of the phys-ical characteristics on the electronic device to achieve the computation process, while digital systems tend to rely on logic gates to perform the computation process. In contrast, the biological brain is an analog system that relies on phys-ical properties for computation rather than Boolean logic. Because the neural network can be resistant to noise and faults, it is a good solution for analog implementation [34]. Two major categories of digital systems are processed to neuromorphic implementation that are FPGA and ASIC. The rst one for FPGA, which has been used frequently in neuromorphic systems [17]. Another one is custom or application-speci c integrated circuit (ASIC) chips, which is also common neuromorphic implementations [35].

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing

For neuromorphic systems, custom analog integrated cir-cuits have several universal features which make them suit-able for each other. There are all properties that occur in both analog circuits and biological systems, such as the con-servation of charge, ampli cation, thresholding and integra-tion [36]. Because of the analog circuits similar to biological systems, widely used in hybrid analog/digital neuromorphic systems for the implementation of neuronal and synaptic components. Moreover, several problems with analog sys-tems about unreliability can be addressed by using digital components. Meanwhile, analog neuromorphic systems of synaptic weights are often stored in digital memories. Neuro-morphic system communication includes both intra-chip and inter-chip communication [37].

One of the software tools on the neuromorphic system includes custom hardware synthesis toolset. These synthesis tools usually require a relatively high level of description and conversion, which can be used to implement a low-level representation of the neural circuits on the neuromorphic sys-tem [38]. The second set of software tools for the neuromotor system is a programming tool of neuromotor systems, which include two functions: mapping and programming [39]. The software simulator developed to test and verify the neu-romotor system that is based on a software-based simula-tor for hardware performance. For applications, in order to demonstrate the computational and device capabilities of neuromorphic computing, various types of neural networks have been applied to different applications area, includ-ing images [40], speech [41], data classi cation [42], con-trol [43], and anomaly detection [44]. To achieve these types of applications on hardware, neural networks matched lower power consumption, faster calculations, and footprint ratios delivered are superior to those delivered by using von Neu-mann architecture.

The rest of the paper is organized as follows: Section II introduces the details of Hop eld algorithm. Section III extends to discrete Hop eld network architecture and hard-ware implementation. Section IV describes the learning method in Hop eld algorithm. Section V presents and com-pares the applications of Hop eld algorithm and shows the application development details. Section VI discusses some of the research open challenges and future trends in the Hop-eld algorithm. Section VII summarizes the entire discussion of hardware research on Hop eld algorithms and hardware implementation.

**II. HOPFIELD ALGORITHM**

The Hop eld network is an important algorithm in the history of neural network development. Hop eld [45], a physicist at the California Institute of Technology, proposed in 1982 that it is a single-layer feedback neural network. Hop eld neural network is the simplest and most applicable model in feed-back networks [46], because it has the function of associative memory [47], which can accurately identify the object and accurately identify digital signals even if they are contami-nated by noise.

VOLUME 8, 2020

The Hop eld neural network model is a kind of recurrent neural network [48]. There is a feedback connection between the input and the output. Under the input excitation, it will be in a constant state of ux. The feedback network can be divided as stable and unstable, which is by judging its stability. For a Hop eld network, the key is to determine its weight coef cient under stable conditions [48]. If the weight matrix W of the Hop eld the network is a symmetric matrix, and the diagonal elements are 0 then it indicates that the network is stable.

According to the discrete or continuous output of the network, the Hop eld network is divided into two types: discrete Hop eld neural network (DHNN) and continuous Hop eld neural network (CHNN) [49]. Discrete Hop eld Neural Network (DHNN): The output of a neuron takes only 1 and 0, which is respectively indicating the neuron in an acti-vation and inhibition state [50]. Continuous Hop eld Neural Network (CHNN) of topology structure is identical to the DHNN. But the difference is whether its activation function is a discrete step function [51] or a continuous function of sigmoid [52].

Due to the structural characteristics of discrete Hop eld network, the output data is equal to the mode size and dimen-sion of the input. Meanwhile, it is the neurons that take binary values (1, 1) or (1, 0) as input and output. The synaptic weight between neuron *i* and neuron *j* is *Wij* [53]. So for a Hop eld neural network with N numbers of neurons, the weight for the matrix is NxN size. Its unique associative memory process is through a series of the iterative process until the system is stable [54].

Discrete Hop eld network is a feature that can be used for associative memory. This is one of the intelligent characteris-tics of human beings, so the Hop eld algorithm can simulate human ‘‘want’’ [55]. By reviewing and thinking about the past scenes, it is used as the associative memory of the Hop eld Neural network. Firstly, learning training process to determine the weight coef cient of the network, and then the memorized information is stored with minimal energy in the N-Dimensional hypercube of a certain corner [56]. Meanwhile, after the weight coef cient of the network is determined, as long as a new input vector is given to the network, this vector may be local data, incomplete or partially incorrect data, but the network still produces a complete output of the information being remembered [57].

The most prominent feature of the Hop eld neural net-work concept is designed closely related to circuit hardware deployment [58]. The main idea of the Hop eld is the use of the hardware circuit to simulate neural network optimiza-tion process. This process can be fast that takes an analog circuit processing advantage rather than digital circuit [59]. Unlike the software realization of the Hop eld neural net-work, the hardware implementation of the algorithm makes brain-like computations possible [60].

Hop eld is based on the idea of energy function to create a new calculation method, which is through the nonlinear dynamics method for developing this neural network. It has

67087

Z. Yu *et al.*: Overview of Neuromorphic Computing

clari ed the relationship between the neural network and dynamics model [61]. Then, established the stability crite-rion of the neural network on this algorithm. Meanwhile, it points out that the information is stored in the connection between the neurons of the network, eventually results build a Hop eld network. In addition, Hop eld algorithm com-pares the feedback network with the Ising model in statistical physics and de nes the upward and downward directions of the magnetic rotation as neuron’s two states of activation and inhibition [62]. That means the magnetic rotation interaction as the synaptic weight on the neuron. This logicality helped many physics theory and physicists to enter the eld of neural networks. In 1984, Hop eld designed and developed the circuit of the network algorithm model [63], it is stating that neurons can be implemented with operational ampli ers, and all neuron connections can be simulated by electronic cir-cuits [64]. One of the continuous Hop eld networks using cir-cuit deployed, which is successfully solved travelling sales-man problem (TSP) calculation problem. It proves that the Hop eld circuit can address the optimization problem [65].

Moreover, Hop eld network can convert analog signals into a digital format that is to realise associative mem-ory, signal estimation and combination optimization appli-cations [66]. This solution is similar to the method of the human rst layer to achieve signal processing. So, it belongs to the neuromorphic calculation. Due to the algorithm sta-bility of output digital signal, Hop eld neural can withstand the redundant input of analog signal noise or variable [67]. This situation is in contrast to the interface circuit between the traditional analog transmission medium and the digital computing device [68]. It takes the speed advantage of the analog circuit and the noise reduction ability of the digital circuit into account.

**III. DISCRETE HOPFIELD NETWORK**

Hop eld algorithm as a single-layer fully interconnected feedback network that includes symmetric synaptic con-nections to stores information on the connections between neurons. It is forming a discrete neural network that is char-acterized by parallel processing [69], fault tolerance and trainability [70].

Discrete Hop eld network of function that simulates the memory of biological neural network is often called asso-ciative memory network. Associative memory (AM) is an integral part of neural network theory [71], and it is a sig-ni cant function in arti cial intelligence and other elds that are used for pattern recognition [70], image restoration [72], intelligent control [73], optimal computation [74] and optical information processing [75]. It mainly uses the good fault tol-erance of neural networks to restore incomplete, defaced and distorted input samples achieve complete prototypes, which are suitable for recognition, classi cation purposes [76].

Association is based on memory where the information is stored rst, and then retrieved in a certain way or rule. Associative memory is also called content-addressed mem-ory, which means the process of associative memory is the

67088

process of accessing information [77]. Information is dis-tributed in the content of biological memory, rather than a speci c address. The storage of information is distributed, not centralized. The storage of information is content address-able memory (CAM) that is distributed, not centralized [78]. Whereas traditional computers are based on addressable memory, which is a group of information with a certain storage unit [79]. In comparison, Hop eld neural networks are more consistent with the information storage mode of biological memory. It is distribution stores the information in the connection weight matrix of the neural network that can be recalled directly from the content of the information [53].

According to different memory recall methods, associative memory networks can be divided into static memory and dynamic memory networks. The former advocated a forward mapping of inputs, while the latter of the memory process is the interactive feedback of input and output. Since the dynamic network has good fault tolerance, it is the most com-monly used associative memory [80]. The common dynamic memory network is the Hop eld model (auto-associative memory) [81] and Kosko’s Bidirectional Associative Mem-ory (BAM) model (hetero-associative memory) [82].

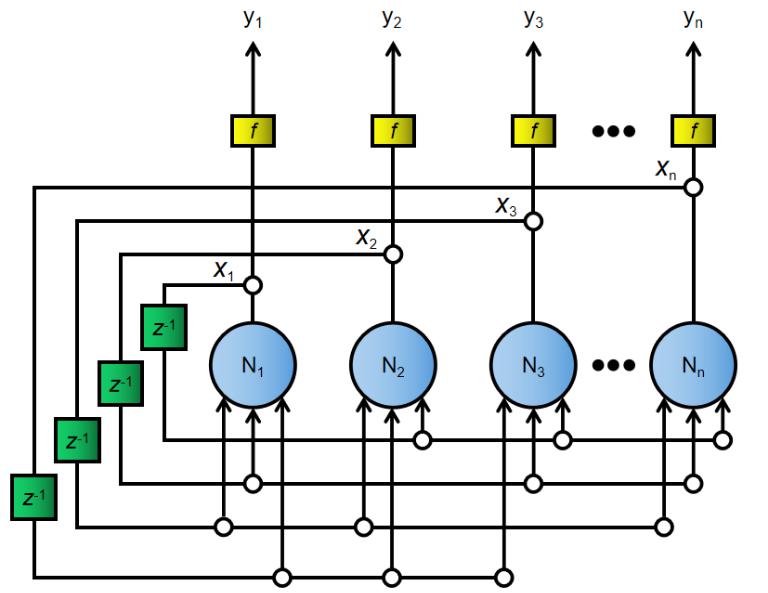
The applied classi cation based on associative mem-ory can be divided into auto-associative memory and hetero-associative memory. Auto-associative memory refers to recovering from the damaged input mode to the complete mode; it can map the input mode in the network to one of the different modes stored in the network. At this point, the associative memory network can not only map the input to the self stored modes, but also have some fault tolerance for the input mode with default or noise [50]. Hetero-associative memory refers to obtaining other relevant patterns from input patterns. When the hetero-associative network is excited by input patterns with certain noise, it can associate the pattern pairs of the original samples through the evolution of the state [83].

In the process of realizing associative memory, discrete Hop eld neural network is divided into two working stages: learning-memory stage and associative memories stage. The task of the learning-memory stage is to adjust the weights based on the input samples, so that the stored samples become dynamics factors [84]. The task of the associative memories stage is to make the nal steady state as attractor dynamics after the weights are adjusted, it is according to the given incomplete or affected information as the associative key-word [85]. In fact, this associative memory process is the con-tinuous movement of the energy function inside the Hop eld neural network, so that the energy is continuously reduced, eventually reaching a minimum value, and in a steady-state process [86].

From the perspective of learning processing, Hop eld algorithm is a powerful learning system with simple structure and easy programming. The Hop eld network operates in a Neural Dynamics, and its working process is the evolution of the state, which means the evolution from the initial state in the direction of energy reduction until it reaches a stable

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing



**FIGURE 1.** The fully connected network architecture of the Hopfield network (adaptedfrom [91]).

state, which is the output results. Therefore, the state of the Hop eld network evolves in the direction of decreas-ing energy function. Since the energy function is bounded, the system will incline to a stable state, which is the output of the Hop eld network [87].

From a system perspective, the feedforward neural net-work model has limited computing power, and the feedback dynamics of a feedback neural network more stronger com-puting power than a feedforward neural network, which is based on feedback to enhance global stability [88]. In feed-back neural networks, all neurons have the same status and there is no hierarchical difference. They can be connected to each other and also feedback signals to themselves [89]. In contrast, although the back-propagation neural network model can handle learning problems, it is not suitable for combinatorial optimization problems. In theory, if the appli-cation is properly set, Hop eld Neural networks can be more robust on the applications. It is a static nonlinear mapping, and the nonlinear processing capability of the complex sys-tem can be obtained by the compound mapping of the simple nonlinear processing unit [90].

The discrete Hop eld neural network (DHNN) which is also known as a feedback neural network is fully con-nected network architecture and is shown in Fig 1. The circles represent neurons, and the output of each neuron as the input of other neurons, which means that the input of each neuron comes from other neurons. In the end, other neurons will return the output data to themselves. At this time, each neuron the input and output has a delay

* 1 [91]. In the Hop eld neural network each neuron is of the same model, and x represents the neuron output at the current time, and y represents the neuron output at the next time.

So, in the time *t*, the output of x in the neuron i can be expressed asV

|  |  |  |
| --- | --- | --- |
| *n* |  |  |
| X | (1) |  |
| *xi* .*t*/D*wijyj* .*t*/ |  |

*j*D*1*

In the time *t*C*1*, the output of *y* in the neuron *i* can be expressed asV

|  |  |
| --- | --- |
| *yi* .*t* C *1*/D *f* .*xi* .*t*// | (2) |

where *f ( )* is the transfer functionV

|  |  |  |
| --- | --- | --- |
| ( | *a 0* |  |
| C1 |  |
| *f* .*a*/D | (3) |  |
| 1 | *a* < *0* |  |

Converting the network structure into a circuit topology that is shown in Fig 2, Hop eld neural networks are equiv-alent to ampli ed electronic circuits. Input data for each electronic component (neuron), including constant external current input, and feedback connections that to link with other electronic components [92].

As shown in Fig 2, each electronic component is based on ampli ers. It includes a non-inverting ampli er and an invert-ing ampli er (depending on the positive and negative weight of the connection to select corresponding output needs) [93]. All states are feedback to the input of the circuit through the bias current *Is* (*S* D *1,2,3,. . . , N*) [94]. At the connection point of each neuron, there is has a resistor, which represents the impedance R*ij*(R*ij* D 1/*Wij*) connected to other neurons [95], and the constant *Wij* represents the network weight between neuron *i* and neuron *j*.

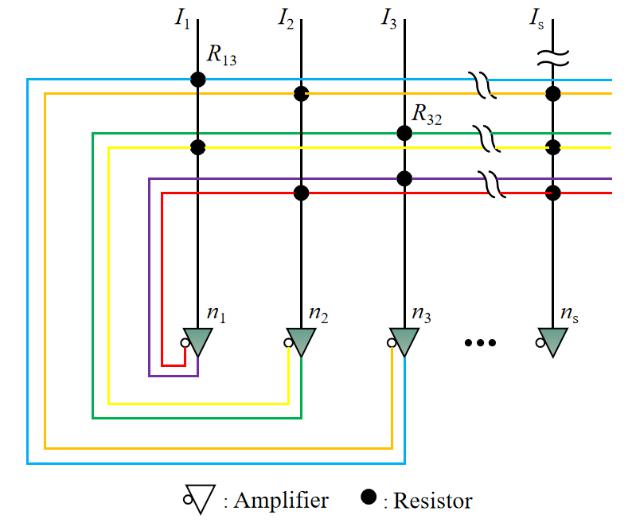
For the bias current calculation is shown in the followingV

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *s* | *xi* | *s* |  |  |  |
| X |  | X |  | (4) |  |
| *Ii* D *j*D1 |  | D *j*D1 | *xjwij* |  |
| *rij* |  |

|  |  |
| --- | --- |
| VOLUME 8, 2020 | 67089 |

Z. Yu *et al.*: Overview of Neuromorphic Computing

**FIGURE 2.** Circuit topology of Hopfield network.



**IV. LEARNING METHOD**

The neural network learning method can be classi ed by event sequence and time series [96]. This occurrence is due to the asynchronous timestamps, where the sequence of events depend on the network dynamics, and the time-series is deterministic [97]. For instance, in the RNN (recurrent neural network) architecture of the Backpropagation Through Time Algorithm (BPTT), Forward propagation is calculated each time in sequence order, while the Backpropagation delivers the accumulated residuals starting from the last time sequence number through the multiple layers [98]. In contrast, the event information can be captured based on the event sequence method, thereby, adjusting the time steps of the conditional intensity function [96]. On the other hand, the Hop eld algo-rithm training is based on the dynamics evolving in discrete time with time steps to discrete learning [99].

According to different learning environments, neural net-work learning methods can be divided into supervised learn-ing and unsupervised learning [100]. In supervised learning, the data of the training samples are loaded to the network input end. Meanwhile, the corresponding expected output is compared with the network results to achieve the difference value [101]. So, it is to adjust the connection strength on the weights and converge to a certain weight after repeated train-ing [102]. If the sample situation is changed, then weights can be modi ed to adapt to the new environment after train-ing [103]. Neural network models using supervised learning include back-propagation networks [104], perceptrons [105], etc. In unsupervised learning, the network is directly placed into the environment without giving standard samples, and the learning and the working combined as a stage [106]. At this point, the learning rule transformation is based on the evolved equation of connection weight [107]. The clas-sic example of unsupervised learning is the Hebb learning rule [108].

Hebb learning rules are the basis of arti cial neural net-works. The adaptability of neural networks is realised through

67090

learning approaches [109]. It is a behaviour according to the environment changes, which is used to adjust weights and then to improve the system [110]. Hebb rules believe that the learning process occurs in the process of synapse between neurons. At the synapse, the strength of synaptic connections varies with the activity of neurons between synapses [111]. Some arti cial neural network learning rules can be regarded as deformation by the Hebb learning rules [112]. Based on this, researchers have proposed various learning rules and algorithms to meet the needs of different network mod-els. Effective learning algorithms enable neural networks to construct different target and object representations, which is through adjustment of connection weights. It is forming distinctive information processing methods that to enabling information storage and processing is re ected in network connections [113].

In 1949, D.O. Hebb proposed the ‘‘synaptic correction’’ on the learning mechanism of neural networks by psychology hypothesis [111]. Its means when neuron i and neuron j are excited at the same time, the connection strength between the two neurons should be enhanced. For example, in animal experiments when a bell rings, a neuron is excited, and at the same time the appearance of food will stimulate the other nearby neurons, then the connection between these two neurons will be strengthened, so that there is a relation-ship between these two things connected. On the contrary, if two neurons are always unable to stimulate simultaneously, the connection between them will become weaker [114].

The neuron stores the learned knowledge on the connection weights of the network. From the biological eld, when the A cell’s neuron axon is close enough to B cells, it repeat-edly and continuously stimulates to cell B. At this point, the connection between the two cells will be strengthened. This means one or two cells in A or B will produce some kind of growth process or metabolic change, thereby enhancing the stimulation effect of cell A into cell B [115].

The Hebb learning rule can be mathematically expressed as followsV

|  |  |
| --- | --- |
| *Wij* .*t* C1/D *Wij* .*t*/C *a yi yj* | (5) |

The *Wij* represents the connection weight of neuron *j* to neuron *i*, *yi* and *yj* represent the output of two neurons, ‘a’ is a constant representing the learning rate. If *yi* and *yj* are activated at the same time, that means *yi* and *yj* are positive, and the *Wij* will increase. If *yi* is activated and *yj* is inhibited, that means *yi* is positive and *yj* is negative, then *Wij* will decrease. This equation shows that the change in weight *Wij* is proportional to the product of the transfer function values on both sides of the synapse. This is an unsupervised learning rule, which does not require any information from object outputs [116].

Hop eld neural network weight learning uses the sum of the outer-products method by Hebb rule. Given P pattern samples (n dimensional orthogonal vector) that is X*p*, P D 1, 2, 3, . . . , P, *x* 2 { 1, 1}*n*, and the samples are orthogonal to

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing

each other. The *n> p*, then the weight matrix is outer product sum of memory samples [117].

Outer product sumV

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *P* |  | *p* |  | *p* | *T* |  |  |  |
|  |  | X | |  |  |  |  |  |
| *W* D | |  |  | *x* |  |  |  | (6) |  |
|  | *x* | |  |  |  |  |  |
|  |  | *P*D1 | |  |  |  |  |  |  |  |  |
| Component-wise mannerV | | | |  |  |  |  |  |  |  |  |
| *Wij* D( | 0; |  | *P* |  | *i* | *j* |  | *i* | 6D*j* | (7) |  |
|  | *P*D1 |  |  |  |
|  | X | |  | *xpxp*; | | | | *i* | *j* |  |  |
|  |  |  |  |  |  |  |  |  |  |

D

At this point, *W* satis es the symmetry requirement, and its need to check whether is an attractor on the algorithm.

Since *P* samples X*p*, *P* D *1, 2, 3, . . . , P*, x 2{ 1, 1}*n* is pair-wise orthogonal. The calculation according to the following equationV

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | D | (*n* | *p* D *k* |  |
| *xp* | *T xk* |  | 0 | *p* 6D*k* | (8) |

Due to *n > p*, the attractor x*p* will be calculated by the

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| following | |  |  |  |  |  |  |  |  |  |
| *f* | *wxp* |  | D *f* | .n-p/ *xp* |  | D *sgn* | .n-p/ *xp* | D *xp* | (9) |  |
| The |  |  |  |  |  |  | net- |  |
|  | weights computing | | | | | work ow of | | Hop eld |  |

work [118] as following:

**Algorithm 1** Hop eld Algorithm Work ow

**Input:** *Pn*pattern samples (*n*dimensional orthogonal vec-tor);

1. Set to the initial state of the network X D P;
2. Set the number of iteration steps;

3: Calculate the *W* weight of the network: *W* D P*i*D1*n* *PT* *P I* ;

1. Since *Wij* D 0, subtract the unit matrix I;
2. Perform iterative calculation;
3. Until the number of iteration steps is reached or the state of the network is stabled, stop the network learning operations, otherwise iteration continues;

**Output:** Weight matrix of Hop eld Neural Network

However, when the network size is xed, the number of memory mode is limited. For the allowed association error rate, the maximum number of memory modes P*max* that the network store ability of the capacity is *N*/log(*N*). It is related to the network size, algorithm and the distribution of vectors in the memory mode [119]. When designing a DHNN net-work using an outer product method, if the memory patterns all meet the pairwise orthogonal condition, the *n*-dimensional network can memorise at most n patterns [120]. Nonethe-less, the pattern samples cannot all meet the orthogonality condition, and the information storage of the network will be greatly reduced for the non-orthogonal patterns. In fact, when the network size *n* is set, the more patterns to be memorised, which cause the high possibility of errors in associations. On the contrary, if it is required the lower error

VOLUME 8, 2020

**FIGURE 3.** Hopfield network of the recognition application workflow.



rate, which needs the smaller information storage capacity of the network. When it exceeds 0.14*n* proportion, an error may occur during the association on the network [121]. The error result corresponds to a local minimum of energy, or a pseudo-attractor [122].

**V. APPLICATIONS**

In daily life, character recognition has high practical appli-cation value in the postal [128], transportation [129] and document management process [130], such as the recog-nition of car numbers and license plates in transportation systems [131]. However, the images captured in the natural environment are often blurred due to the limitations of camera hardware [111], or uncleared by the font is occluded and worn out. At these points, the complete information of the char-acter cannot be obtain and identi cation of noisy characters become a key issue [132].

At present, there are several methods for character recog-nition, which are mainly divided into a neural network [133], probability statistical [134], and fuzzy recognition [135]. The traditional character recognition method cannot recog-nize well under the condition of noise interference. How-ever, the discrete Hop eld neural network has the function of associative memory, which is reasonable for anti-noise processing [136]. By applying this function, characters can be recognized and satisfactory recognition results can be achieved. Besides, the convergence calculation becomes fast for processing.

**A. CASE STUDY: CHARACTER RECOGNITION**

The associative memory can be designed based on the dis-crete Hop eld neural network concept, and the network can recognize the 10 digits that fall in the range from 0-9. Fur-thermore, despite any disturbance by certain noise due to the speci ed range of numbers, still has a good recognition effect by network feedback. At this point, the network is composed of a total of 10 stable states that reach to 0-9 numbers. These numbers are represented by 10 10 matrices and are directly described by the binary matrix. In the 10 10 matrix, the number pixel is represented by 1, and the non-number pixel is de ned 1 as blank display.

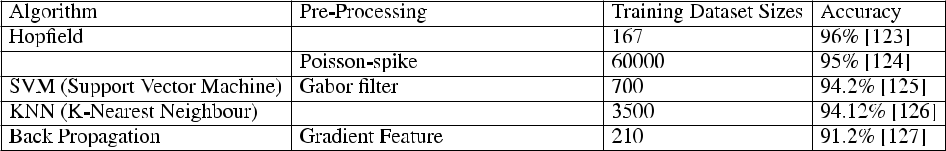
The network through learning the above matrices on the function of associative memory is performed to achieve 10 steady states reach 10 numbers. When noisy data are applied to the network, the output of the network which is a feedback, used to determine the comparison of 10 steady states such as the object vector. Finally, the purpose of the whole operation is to achieve the effect of correct recognition.

The Hop eld network of recognition application work ow is depicted as below:

67091

Z. Yu *et al.*: Overview of Neuromorphic Computing

**TABLE 1.** Algorithm comparison on the character recognition cases.



1) STANDARD SAMPLES

In this study, standard samples are selected to convert into binarisation matrix. In this application of characters recog-nition, all the numbers from 0-9 are converted into 10 10 matrices.

2) DESIGN HOPFIELD NETWORK

In this case, network T is a matrix of R\*Q with Q target vectors (the element must be binary of 1 or 1). According to the above matrices requirements that to design a network architecture, it is a 10 10 matrix on number sample size. At this point, the Hop eld network should contain 100 neu-rons, which reach input a 10 10 matrix into an algorithm.

3) HOPFIELD NETWORK TRAINING

The learning processing adopts a neurodynamic method. The working process is based on the evolution of the state. For a given initial state, it evolves of energy decreasing manner, and nally reaches a stable state. The network starts from the initial state X (0) to multiple recursions, where its state does not change to form a stable state, which means the network is stable by X (t C 1) D X (t).

When the X state is reached to a steady-state, then it is considered as an attractor. For the attractors, it is determined of nal behaviour on a dynamic system. The system require-ment of remembers information that is stored in different attractors. When the input sample containing part of the memoried information applied to a network, the evolution process of the network is resumed all the required sample information from the inputted part of the information. The aforementioned procedure is called the process of associative memories.

1. INPUT TEST SAMPLE TO HOPFIELD NEURAL NETWORK

In this section, test samples are converted to the matrix, and the size is the same as the training sample. This new test matrix is then placed into a trained Hop eld network, and the network output of the feedback, which is found to the closest of the object vector. In this case, the feedback will be in a range from 0-9 matrices.

1. RESULT ANALYSIS

The test results are analyzed and compared with the trained object vector. Through a con dence level calcula-tion, the resulting output the closest of the object vector can be obtained. Finally, the classi cation result is achieved. Further research shows the recognition effect decreases by the increase of noise intensity. When the noise intensity

67092

exceeds 30%, the Hop eld network hardly recognizes the number [137], and fail to remember correct object when the noise intensity destroy the input over 50% [138].

The table 1 has shown some similar approaches with Hop-eld algorithm. And compared accuracy with in the charac-ter recognition application. The Hop eld algorithm is most dataset request and best accuracy on the application, which is means the Hop eld algorithm is suitable on this character recognition eld.

**B. OTHER APPLICATIONS**

Based on the discrete Hop eld neural network, it has the function of associative memory. In recent years, many researchers have attempted to apply Hop eld network to various elds in order to replace conventional techniques to address the issues, such as water quality evaluation [139] and generator fault diagnosis [147], and have achieved consider-able results by applying aforementioned method. For exam-ple, in the Internet of Things (IoT) applications, where mul-tiple links fail and break the real-time transmission services, and due to this reason, the fault cannot be quickly located at that particular point [148]. The relationship between the fault set and the alarm set can be established through the network topology information and the transmission service, which is compatible with the proposed Hop eld Neural Net-work [149]. The built-in Hop eld algorithm of the energy function is used to resolve fault location, and hence, it is found that integration of aforementioned algorithm with the IoT will improve transmission services in smart cities [150].

However, the application will have a wider framework to suitable more applications in addition to limited applicabil-ity for the eld. When the Hop eld neural network collab-orates with some notable optimisation algorithms, not the network alone, that provides its associative memory stronger but also improves the application ef ciency. For example, the existence of many pseudo-stable points in general discrete Hop eld neural networks, limit the pro ciency of network. Therefore, a Genetic algorithm can be considered for the dis-crete Hop eld network [151], and the global search capability of the Genetic algorithm is used to optimise the steady-state of Hop eld associative memory function. So that makes the associative mode jump out of the pseudo-stable point, and then the Hop eld network maintains a high associative suc-cess rate under the condition of higher noise to signal ratio.

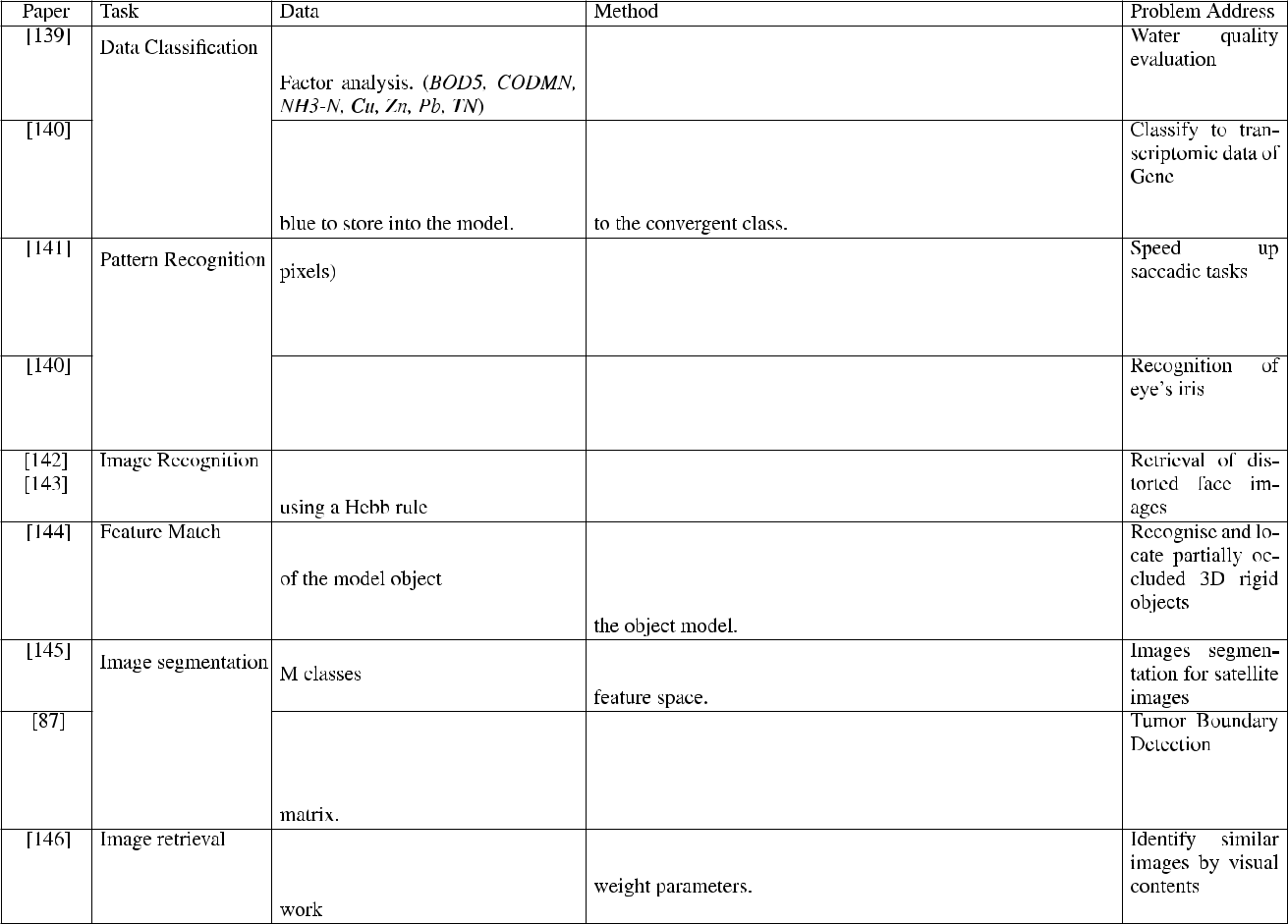
**C. APPLICATION COMPARISON**

The work ow details of Hop eld algorithm are similar to those of above in section B mentioned. However, the main

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing

**TABLE 2.** Algorithm extraction and use cases for Hopfield.



difference is that varied projects need to design suitable tem-plates for different objects to matching data-sets, and then to input object data into the neural network algorithm for learn-ing. Appropriate templates will help the algorithm learning features more easily and improve processing accuracy.

The actual practice of Hop eld neural network involves a surprising number of scienti c disciplines, which is the key technical areas it covers include Data Classi cation, Pattern Recognition, Image Recognition, Feature Match, Image seg-mentation, Image retrieval. The result of the investigation was reported as shown in Table 2. By comparing the applications in various elds, we can nd that the application of the algorithm is a different implementation method. It is mainly depending on the design of the neural network weights that the algorithm uses the weights for associative memory to output results. Under the appropriate template of input data, which to corresponding output analysed and predicted results.

**VI. FUTURE PLAN AND CHALLENGES**

For the future of neuromorphic chips, it is the key to break through the development direction of von Neumann’s structure limitations. Because the basic operations of neural networks are the processing of neurons and synapses [152],

VOLUME 8, 2020

the conventional processor instruction set (including 86 and ARM, etc.) was developed for general-purpose com-puting [153]. These operations are arithmetic operations (addition, subtraction, multiplication and division) and log-ical operations (AND-OR-NOT) [154]. It often requires hun-dreds or thousands of instructions to complete the processing of neuron computing, making the low processing ef ciency of the hardware inef cient.

Currently, neural computing needs a completely different design than the von Neumann architecture [2]. The storage and processing are integrated into the neural network [11], whereas in von Neumann’s structure, there it is separated and realized respectively by memory and computational unit [155]. There is a huge difference between the two com-puting when using current classical computers based on the von Neumann architecture (such as CPUs and GPUs) to run neural network applications. They are inevitably restricted by a separate storage and handling structure, which has caused a lower ef ciency over the impacts. Although the current FPGA and ASIC can meet the requirements of some neural network application scenarios, a new generation of architecture like neuromorphic chips and integrated com-puting design will be used as the underlying architecture

67093

Z. Yu *et al.*: Overview of Neuromorphic Computing

to improve the neural network computing in the long-term planning [156].

Among the above, ASIC is a kind of chip speci cally designed for this special purpose. Compared to FPGA, it fea-tures stronger performance, smaller size, less power con-sumption, lower cost and more progress in developing hard-ware design. ASIC needs research, development time and high risks of technology marketing that have become a major obstacle to future promotion. However, some of its advantages such as good mold size, low cost, great energy consumption, great reliability, strong con dentiality, high computing performance and high computing ef ciency have become the best choice for current formal nerve chips [157].

Another key point of neural computing is the challenge of holding computing nodes [158]. Generally, the nodes of bit computation are the conduction switches of the tran-sistors [159]. However, formal neural computing requires computational nodes like neurons, which is the penalty for an alternative generalized alternative approach to achieve non-bit computing. This means that arti cial synapses and excitement need improvement [160]. Nowadays, there are a lot of explorations on how to simulate or create syn-thetic synapses. Taken as a whole, for produced formal neu-ronal chips, industrial circuits are used primarily to simulate synapses that achieve formal neuronal computing [4]. But manufacturing processes and technical costs are high and production ef ciency is low, causing neuronal simulation ef ciency to low.

There are still many problems in the research of new materials for the neuromorphic hardware. In the future, researchers in the neuromorphic disciplines consider new materials belonging to neuromorphic computing can be found in place of transistors to new hardware design [161]. For example, the array composed of memristor that is a plas-tic element can be stored and processed to integrate for the neuromorphic hardware. It has a high switching current ratio, a light effective mass, a large adjustable band gap and large electron mobility, which provides a favourable basis for successful preparation of low-power neuromorphic hard-ware [162].

Eventually, the architecture, algorithm and programming scheme of adaptive neuromorphic computing is in a wide blank and a long way to reach a nal goal that replaces to von Neumann’s structure in the arti cial intelligence discipline. But the frontiers of neuromorphic computing knowledge are being pushed farther outwards over the time, and the future opens a bright prospects.

**VII. CONCLUSION**

Although neuromorphic computing has gained widespread attention in recent years, however, it is still considered to be in the infancy stage. The existing solutions mostly focus on a single application at the hardware or software level, and majority of them are only suitable for handling lim-ited applications. In addition, there are many software-based neural network applications that has been deployed, but

67094

hardware-based neural network design has been the key to the neuromorphic design. Convention neural network circuit implementation is thought of time-consuming and inconve-nient. In order to apply a simple and fast design method to neural network hardware, which can optimise and manufac-ture neuromorphic computing systems, needs to systemati-cally uni cate the requirements of the software calculation process. Furthermore, it can process and improves the nal software-level application indicators to quantify hardware attributes. Finally,a testable solution for a speci cation com-ponent can be achieved.

In this study, we have attempted to give an overview of work that has been carried out in the hardware implemen-tation eld on neural networks. In addition, we have also discussed the various techniques and methods employed in the overall progression and implementation of Hope eld Algorithm. In this regard, it is found that this algorithm has been extensively deployed in various disciplines based on feasibility and ef ciency. Moreover, we have also high-lighted the existing solutions for neuromorphic computing which are mainly focused on a single application at the software-hardware level. In this regard, it is discovered that there is signi cant room for further improvement to achieve the most optimized design with a low computation process. From in-depth research, we strongly believe that this paper could provide a signi cant step towards the hardware imple-mentation of low-power neuromorphic processing systems using advanced Hop eld algorithm.

**ACKNOWLEDGMENT**

Authors would like to thank Sultan Qaboos University (Government of the Sultanate of Oman) for supporting Dr. A. M. Abdulghani.

**REFERENCES**

1. C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean,
   1. S. Rose, and J. S. Plank, ‘‘A survey of neuromorphic computing and neural networks in hardware,’’ 2017, *arXiv:1705.06963*. [Online]. Available: http://arxiv.org/abs/1705.06963
2. I. K. Schuller, R. Stevens, R. Pino, and M. Pechan, ‘‘Neuro-morphic computing From materials research to systems architec-ture roundtable,’’ USDOE Of ce Sci. (SC), Washington, DC, USA, Tech. Rep. Report of a Roundtable Convened to Consider Neuromorphic Computing Basic Research Needs, 2015.
3. D. Monroe, ‘‘Neuromorphic computing gets ready for the (really) big time,’’ *Commun. ACM*, vol. 57, no. 6, pp. 13 15, 2014.
4. B. Rajendran, A. Sebastian, M. Schmuker, N. Srinivasa, and
   1. Eleftheriou, ‘‘Low-power neuromorphic hardware for signal processing applications: A review of architectural and system-level design approaches,’’ *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 97 110, Nov. 2019.
5. P. Hasler and L. Akers, ‘‘VLSI neural systems and circuits,’’ in *Proc. 9th* *Annu. Int. Phoenix Conf. Comput. Commun.*, 1990, pp. 31 37.
6. J.-C. Lee and B. J. Sheu, ‘‘Parallel digital image restoration using adaptive VLSI neural chips,’’ in *Proc. Int. Conf. Comput. Design, VLSI Comput.* *Process.*, 1990, pp. 126 129.
7. L. Tarassenko, M. Brownlow, G. Marshall, J. Tombs, and A. Murray, ‘‘Real-time autonomous robot navigation using VLSI neural networks,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 1991, pp. 422 428.
8. M. Davies *et al.*, ‘‘Loihi: A neuromorphic manycore processor with on-chip learning,’’ *IEEE Micro*, vol. 38, no. 1, pp. 82 99, Jan. 2018.
9. S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. Hoboken, NJ, USA: Wiley, 2014.

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing

1. S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, ‘‘A scalable multicore architecture with heterogeneous memory structures for dynamic neu-romorphic asynchronous processors (DYNAPs),’’ *IEEE Trans. Biomed.* *Circuits Syst.*, vol. 12, no. 1, pp. 106 122, Feb. 2018.
2. G. Indiveri and S.-C. Liu, ‘‘Memory and information processing in neuromorphic systems,’’ *Proc. IEEE*, vol. 103, no. 8, pp. 1379 1397, Aug. 2015.
3. A. L. Hodgkin and A. F. Huxley, ‘‘A quantitative description of mem-brane current and its application to conduction and excitation in nerve,’’
   1. *Physiol.*, vol. 117, no. 4, pp. 500 544, 1952.
4. Q. Ma, M. R. Haider, V. L. Shrestha, and Y. Massoud, ‘‘Burst-ing Hodgkin Huxley model-based ultra-low-power neuromimetic sil-icon neuron,’’ *Anal. Integr. Circuits Signal Process.*, vol. 73, no. 1, pp. 329 337, Oct. 2012.
5. A. Borisyuk, *Morris-Lecar Model*. New York, NY, USA: Springer, 2015,

pp. 1758 1764, doi: [10.1007/978-1-4614-7320-6\_150-1.](http://dx.doi.org/10.1007/978-1-4614-7320-6_150-1)

1. M. Graupner and N. Brunel, ‘‘Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic loca-tion,’’ *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 10, pp. 3991 3996, Mar. 2012.
2. A. Galves and E. Löcherbach, ‘‘In nite systems of interacting chains with memory of variable length A stochastic model for biological neural nets,’’ *J. Stat. Phys.*, vol. 151, no. 5, pp. 896 921, 2013.
3. F. Grassia, T. Levi, T. Kohno, and S. Saïghi, ‘‘Silicon neuron: Digital hard-ware implementation of the quartic model,’’ *Artif. Life Robot.*, vol. 19, no. 3, pp. 215 219, Nov. 2014.
4. E. M. Izhikevich, ‘‘Which model to use for cortical spiking neurons?’’ *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063 1070, Sep. 2004.
5. M. Hayati, M. Nouri, D. Abbott, and S. Haghiri, ‘‘Digital multiplierless realization of two-coupled biological Hindmarsh Rose neuron model,’’ *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 5, pp. 463 467,May 2016.
6. J. V. Arthur and K. Boahen, ‘‘Silicon neurons that inhibit to synchronize,’’ in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, p. 4.
7. W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons,*

*Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

1. W. S. McCulloch and W. Pitts, ‘‘A logical calculus of the ideas immanent in nervous activity,’’ *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115 133, 1943.
2. A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and
   1. Maida, ‘‘Deep learning in spiking neural networks,’’ *Neural Netw.*, vol. 111, pp. 47 63, Mar. 2019.
3. J. H. Kotaleski and K. T. Blackwell, ‘‘Modelling the molecular mecha-nisms of synaptic plasticity using systems biology approaches,’’ *Nature* *Rev. Neurosci.*, vol. 11, no. 4, pp. 239 251, Apr. 2010.
4. E. Neftci, ‘‘Stochastic neuromorphic learning machines for weakly labeled data,’’ in *Proc. IEEE 34th Int. Conf. Comput. Design (ICCD)*, Oct. 2016, pp. 670 673.
5. E. Donati, F. Corradi, C. Stefanini, and G. Indiveri, ‘‘A spiking imple-mentation of the lamprey’s central pattern generator in neuromorphic VLSI,’’ in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, pp. 512 515.
6. L. O. Chua and L. Yang, ‘‘Cellular neural networks: Applications,’’ *IEEE* *Trans. Circuits Syst.*, vol. CAS-35, no. 10, pp. 1273 1290, Oct. 1988.
7. Z. Wang, Y. Ma, F. Cheng, and L. Yang, ‘‘Review of pulse-coupled neural networks,’’ *Image Vis. Comput.*, vol. 28, no. 1, pp. 5 13, 2010.
8. J. Secco, M. Farina, D. Demarchi, and F. Corinto, ‘‘Memristor cellular automata through belief propagation inspired algorithm,’’ in *Proc. Int.* *SoC Design Conf. (ISOCC)*, Nov. 2015, pp. 211 212.
9. Y.-H. Kuo, C.-I. Kao, and J.-J. Chen, ‘‘A fuzzy neural network model and its hardware implementation,’’ *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 3, pp. 171 183, Aug. 1993.
10. Z. Uykan, ‘‘Continuous-time hop eld neural network-based optimized solution to 2-channel allocation problem,’’ *Turkish J. Electr. Eng. Com-put. Sci.*, vol. 23, no. 2, pp. 480 490, 2015.
11. R. T. Cabeza, E. B. Viciedo, A. Prieto-Moreno, and V. M. Vega, ‘‘Fault diagnosis with missing data based on hop eld neural networks,’’ in *Mathematical Modeling and Computational Intelligence in Engineering Applications*. Cham, Switzerland: Springer, 2016, pp. 37 46.
12. X. Hu and T. Wang, ‘‘Training the hop eld neural network for classi - cation using a STDP-like rule,’’ in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 737 744.
13. G. Indiveri, ‘‘Computation in neuromorphic analog VLSI systems,’’ in *Neural Nets WIRN Vietri-01*. London, U.K.: Springer, 2002, pp. 3 20.

VOLUME 8, 2020

1. F. Blayo and P. Hurat, ‘‘A VLSI systolic array dedicated to hop eld neural network,’’ in *VLSI for Arti cial Intelligence*. Boston, MA, USA: Springer, 1989, pp. 255 264.
2. C. Mead, ‘‘Neuromorphic electronic systems,’’ *Proc. IEEE*, vol. 78, no. 10, pp. 1629 1636, Oct. 1990.
3. J. V. Arthur and K. Boahen, ‘‘Learning in silicon: Timing is everything,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 75 82.
4. . Bayraktaro§lu, A. S. Ö§renci, G. Dündar, S. Balk r, and E. Alpayd n, ‘‘ANNSyS: An analog neural network synthesis system,’’ *Neural Netw.*, vol. 12, no. 2, pp. 325 338, Mar. 1999.
5. D. Brüderle *et al.*, ‘‘A comprehensive work ow for general-purpose neu-ral modeling with highly con gurable neuromorphic hardware systems,’’ *Biol. Cybern.*, vol. 104, nos. 4 5, pp. 263 296, May 2011.
6. P. Krzysteczko, J. Münchenberger, M. Schäfers, G. Reiss, and A. Thomas, ‘‘The memristive magnetic tunnel junction as a nanoscopic synapse-neuron system,’’ *Adv. Mater.*, vol. 24, no. 6, pp. 762 766, Feb. 2012.
7. Y. Li, Y. Zhong, L. Xu, J. Zhang, X. Xu, H. Sun, and X. Miao, ‘‘Ultrafast synaptic events in a chalcogenide memristor,’’ *Sci. Rep.*, vol. 3, no. 1, Dec. 2013, Art. no. 1619.
8. Y. Li, Y. Zhong, J. Zhang, L. Xu, Q. Wang, H. Sun, H. Tong, X. Cheng, and X. Miao, ‘‘Activity-dependent synaptic plasticity of a chalcogenide electronic synapse for neuromorphic systems,’’ *Sci. Rep.*, vol. 4, no. 1, May 2015, Art. no. 4906.
9. J. Tranchant, E. Janod, B. Corraze, P. Stoliar, M. Rozenberg, M.-P. Besland, and L. Cario, ‘‘Control of resistive switching in AM4Q8narrow gap mott insulators: A rst step towards neuromor-phic applications,’’ *Phys. Status Solidi A*, vol. 212, no. 2, pp. 239 244, Feb. 2015.
10. Y. Chen, G. Liu, C. Wang, W. Zhang, R.-W. Li, and L. Wang, ‘‘Poly-mer memristor for information storage and neuromorphic applications,’’ *Mater. Horizons*, vol. 1, no. 5, pp. 489 506, 2014.
11. J. J. Hop eld, ‘‘Neural networks and physical systems with emergent collective computational abilities,’’ *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554 2558, Apr. 1982.
12. X. S. Zhang, *Neural Networks in Optimization*, vol. 46. Boston, MA, USA: Springer, 2013.
13. S. G. Hu, Y. Liu, Z. Liu, T. P. Chen, J. J. Wang, Q. Yu, L. J. Deng, Y. Yin, and S. Hosaka, ‘‘Associative memory realized by a recon gurable memristive hop eld neural network,’’ *Nature Commun.*, vol. 6, no. 1, pp. 1 8, Nov. 2015.
14. J. Schmidhuber, ‘‘Habilitation thesis: System modeling and optimiza-tion,’’ Page 150 ff Demonstrates Credit Assignment Across Equivalent 1,200 Layers Unfolded RNN, Tech. Rep. 15-04-1993, 1993.
15. R. Ma, Y. Xie, S. Zhang, and W. Liu, ‘‘Convergence of discrete delayed hop eld neural networks,’’ *Comput. Math. Appl.*, vol. 57, nos. 11 12, pp. 1869 1876, Jun. 2009.
16. C. Ramya, G. Kavitha, and D. K. S. Shreedhara, ‘‘Recalling of images using hop eld neural network model,’’ 2011, *arXiv:1105.0332*. [Online]. Available: http://arxiv.org/abs/1105.0332
17. A. A. Adly and S. K. Abd-El-Ha z, ‘‘Ef cient vector hysteresis modeling using rotationally coupled step functions,’’ *Phys. B, Condens. Matter*, vol. 407, no. 9, pp. 1350 1353, May 2012.
18. U.-P. Wen, K.-M. Lan, and H.-S. Shih, ‘‘A review of hop eld neural networks for solving mathematical programming problems,’’ *Eur. J. Oper.* *Res.*, vol. 198, no. 3, pp. 675 687, Nov. 2009.
19. S. Kumar and M. P. Singh, ‘‘Pattern recall analysis of the hop eld neural network with a genetic algorithm,’’ *Comput. Math. Appl.*, vol. 60, no. 4, pp. 1049 1057, Aug. 2010.
20. J. D. Keeler, ‘‘Comparison between Kanerva’s SDM and hop eld-type neural networks,’’ *Cognit. Sci.*, vol. 12, no. 3, pp. 299 329, Jul. 1988.
21. M. A. Akra, ‘‘On the analysis of the Hop eld network: A geometric approach,’’ Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1988.
22. B. Müller, J. Reinhardt, and M. T. Strickland, *Neural Networks: An Intro-*

*duction*. Berlin, Germany: Springer, 2012.

1. J. A. Anderson, *An Introduction to Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.
2. H. K. Sulehria and Y. Zhang, ‘‘Study on the capacity of hop eld neural networks,’’ *Inf. Technol. J.*, vol. 7, no. 4, pp. 684 688, 2008.
3. A. Tankimanova and A. P. James, ‘‘Neural network-based analog-to-digital converters,’’ in *Memristor and Memristive Neural Networks*. London, U.K.: Intechopen, 2018, p. 297.
4. A. H. Marblestone, G. Wayne, and K. P. Kording, ‘‘Toward an integration of deep learning and neuroscience,’’ *Frontiers Comput. Neurosci.*, vol. 10, p. 94, Sep. 2016.

67095

Z. Yu *et al.*: Overview of Neuromorphic Computing

1. G. A. Elnashar, ‘‘Dynamical nonlinear neural networks with perturbations modeling and global robust stability analysis,’’ *Int. J. Comput. Appl.*, vol. 85, no. 15, pp. 14 21, 2014.
2. S. Cocco, R. Monasson, L. Posani, S. Rosay, and J. Tubiana, ‘‘Statisti-cal physics and representations in real and arti cial neural networks,’’ *Phys. A, Stat. Mech. Appl.*, vol. 504, pp. 45 76, Aug. 2018.
3. J. J. Hop eld and D. W. Tank, ‘‘Computing with neural circuits: A model,’’ *Science*, vol. 233, no. 4764, pp. 625 633, Aug. 1986.
4. D. Tank and J. Hop eld, ‘‘Simple ‘neural’ optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit,’’ *IEEE Trans. Circuits Syst.*, vol. CAS-33, no. 5, pp. 533 541, May 1986.
5. J. J. Hop eld and D. W. Tank, ‘‘‘Neural’ computation of decisions in optimization problems,’’ *Biol. Cybern.*, vol. 52, no. 3, pp. 141 152, 1985.
6. A. Tankimanova, A. K. Maan, and A. P. James, ‘‘Level-shifted neural encoded analog-to-digital converter,’’ in *Proc. 24th IEEE Int. Conf. Elec-tron., Circuits Syst. (ICECS)*, Dec. 2017, pp. 377 380.
7. M. S. Badie and O. K. Ersoy, ‘‘A multistage approach to the Hop eld model for bi-level image restoration,’’ Purdue Univ., West Lafayette, IN, USA, ECE Tech. Rep. TR-ECE 95-22, 1995, Paper 143.
8. S. Draghici, ‘‘Neural networks in analog hardware Design and imple-mentation issues,’’ *Int. J. Neural Syst.*, vol. 10, no. 1, pp. 19 42, 2000.
9. Y. Zhu and Z. Yan, ‘‘Computerized tumor boundary detection using a hop eld neural network,’’ *IEEE Trans. Med. Imag.*, vol. 16, no. 1,
   1. 55 67, Feb. 1997.
10. D.-Q. Wang and Z.-J. Li, ‘‘Pattern recogition based on hop eld neural net-work,’’ *J. Wuhan Yejin Univ. Sci. Technol.*, vol. 4, pp. 231 236, Apr. 2005.
11. G. Palm, ‘‘Neural associative memories and sparse coding,’’ *Neural* *Netw.*, vol. 37, pp. 165 171, Jan. 2013.
12. J. K. Paik and A. K. Katsaggelos, ‘‘Image restoration using a modi ed hop eld network,’’ *IEEE Trans. Image Process.*, vol. 1, no. 1, pp. 49 63, Jan. 1992.
13. B. Bavarian, ‘‘Introduction to neural networks for intelligent control,’’ *IEEE Control Syst. Mag.*, vol. 8, no. 2, pp. 3 7, Apr. 1988.
14. S. Matsuda, ‘‘‘Optimal’ Hop eld network for combinatorial optimization with linear cost function,’’ *IEEE Trans. Neural Netw.*, vol. 9, no. 6,
    1. 1319 1330, Nov. 1998.
15. J.-S. Jang, S.-W. Jung, S.-Y. Lee, and S.-Y. Shin, ‘‘Optical implementation of the Hop eld model for two-dimensional associative memory,’’ *Opt.* *Lett.*, vol. 13, no. 3, pp. 248 250, Mar. 1988.
16. J. A. Clemente, W. Mansour, R. Ayoubi, F. Serrano, H. Mecha, H. Ziade, W. El Falou, and R. Velazco, ‘‘Hardware implementation of a fault-tolerant Hop eld Neural Network on FPGAs,’’ in *Neurocomputing*, vol. 171. Amsterdam, The Netherlands: Elsevier, 2016, pp. 1606 1609.
17. S. S. Yau and H. Fung, ‘‘Associative processor architecture A survey,’’ *ACM Comput. Surv.*, vol. 9, no. 1, pp. 3 27, 1977.
18. M. S. Riazi, M. Samragh, and F. Koushanfar, ‘‘CAMsure: Secure content-addressable memory for approximate search,’’ *ACM Trans. Embedded* *Comput. Syst.*, vol. 16, no. 5s, pp. 1 20, Oct. 2017.
19. L. D. Pyeatt, *Modern Assembly Language Programming With the ARM* *Processor*. London, U.K.: Newnes, 2016.
20. M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks:* *From Fundamentals to Advanced Theory*. Hoboken, NJ, USA: Wiley,2004.
21. Y. P. Singh, A. Khare, and A. Gupta, ‘‘Analysis of Hop eld autoassocia-tive memory in the character recognition,’’ *Int. J. Comput. Sci. Eng.*, vol. 2, no. 3, pp. 500 503, 2010.
22. X. Zhuang, Y. Huang, and S.-S. Chen, ‘‘Better learning for bidirec-tional associative memory,’’ *Neural Netw.*, vol. 6, no. 8, pp. 1131 1146, Jan. 1993.
23. H. Jaeger, ‘‘Using conceptors to manage neural long-term memories for temporal patterns,’’ *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 387 429, 2017.
24. B. Ripley, ‘‘Linear discriminant analysis,’’ in *Pattern Recognition and* *Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2007,
    1. 91 120.
25. W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal* *Dynamics: From Single Neurons to Networks and Models of Cognition*.Cambridge, U.K.: Cambridge Univ. Press, 2014.
26. F. A. Unal, ‘‘Temporal pattern matching using an arti cial neural networs,’’ in *Neural Networks and Pattern Recognition*. Amsterdam, The Netherlands: Elsevier, 1998, pp. 77 104.
27. A. Meyer-Baese and V. J. Schmid, *Pattern Recognition and Signal* *Analysis in Medical Imaging*. Amsterdam, The Netherlands: Elsevier,2014.

67096

1. F. Mastrogiuseppe and S. Ostojic, ‘‘A geometrical analysis of global stability in trained feedback networks,’’ *Neural Comput.*, vol. 31, no. 6,
   1. 1139 1182, Jun. 2019.
2. J. F. Pagel and P. Kirshtein, *Machine Dreaming and Consciousness*. New York, NY, USA: Academic, 2017.
3. G. Serpen, ‘‘Hop eld network as static optimizer: Learning the weights and eliminating the guesswork,’’ *Neural Process. Lett.*, vol. 27, no. 1,
   1. 1 15, Feb. 2008.
4. L. Rong and Q. Junfei, ‘‘A new water quality evaluation model based on simpli ed hop eld neural network,’’ in *Proc. 34th Chin. Control Conf.* *(CCC)*, Jul. 2015, pp. 3530 3535.
5. H. Yang and Z. Liu, ‘‘An optimization routing protocol for FANETs,’’ *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1 8,Dec. 2019.
6. J. J. Hop eld, ‘‘Neurons with graded response have collective computa-tional properties like those of two-state neurons,’’ *Proc. Nat. Acad. Sci.* *USA*, vol. 81, no. 10, pp. 3088 3092, May 1984.
7. C. V. Negoita, *Cybernetics and Applied Systems*. Boca Raton, FL, USA: CRC Press, 1992.
8. M. S. Ansari, ‘‘Voltage-mode neural network for the solution of linear equations,’’ in *Non-Linear Feedback Neural Networks*. New Delhi, India: Springer, 2014, pp. 55 104.
9. S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, ‘‘Joint modeling of event sequence and time series with attentional twin recur-rent neural networks,’’ 2017, *arXiv:1703.08524*. [Online]. Available: http://arxiv.org/abs/1703.08524
10. D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time* *Series Analysis and Forecasting*. Hoboken, NJ, USA: Wiley, 2015.
11. S. Roa and F. Nino, ‘‘Classi cation of natural language sentences using neural networks,’’ in *Proc. Flairs Conf.*, 2003, pp. 444 449.
12. P. Zheng, J. Zhang, and W. Tang, ‘‘Analysis and design of asymmetric hop eld networks with discrete-time dynamics,’’ *Biol. Cybern.*, vol. 103, no. 1, pp. 79 85, Jul. 2010.
13. J. Torres, Ed., *First Contact With Deep Learning: Practical introduc-tion With Keras* (WATCH THIS SPACE collection Barcelona Book 5).Barcelona, Spain: Independently Published, 2018.
14. R. Reed and R. J. MarksII, *Neural Smithing: Supervised Learning in*

*Feedforward Arti cial Neural Networks*. Cambridge, MA, USA: MITPress, 1999.

1. M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, ‘‘Arti - cial neural networks-based machine learning for wireless networks: A tutorial,’’ 2017, *arXiv:1710.02913*. [Online]. Available: http://arxiv. org/abs/1710.02913
2. D. C. Park, M. A. El-Sharkawi, and R. J. Marks, ‘‘An adaptively trained neural network,’’ *IEEE Trans. Neural Netw.*, vol. 2, no. 3, pp. 334 345, May 1991.
3. X. Wu, J. Ghaboussi, and J. H. Garrett, ‘‘Use of neural networks in detec-tion of structural damage,’’ *Comput. Struct.*, vol. 42, no. 4, pp. 649 659, Feb. 1992.
4. M. Riedmiller, ‘‘Advanced supervised learning in multi-layer perceptrons From backpropagation to adaptive learning algorithms,’’ *Comput. Standards Interfaces*, vol. 16, no. 3, pp. 265 278, 1994.
5. S. Marinai and H. Fujisawa, *Machine Learning in Document Analysis and* *Recognition*, vol. 90. Berlin, Germany: Springer, 2007.
6. G. Zhang, *Star Identi cation: Methods, Techniques and Algorithms*.

Berlin, Germany: Springer, 2019.

1. T. D. Sanger, ‘‘Optimal unsupervised learning in a single-layer linear feedforward neural network,’’ *Neural Netw.*, vol. 2, no. 6, pp. 459 473, Jan. 1989.
2. A. Yaman, G. Iacca, D. C. Mocanu, M. Coler, G. Fletcher, and M. Pechenizkiy, ‘‘Evolving plasticity for autonomous learning under changing environmental conditions,’’ 2019, *arXiv:1904.01709*. [Online]. Available: http://arxiv.org/abs/1904.01709
3. J. L. McClelland, ‘‘How far can you go with hebbian learning, and when does it lead you astray,’’ in *Processes of Change in Brain and Cognitive* *Development: Attention and Performance XXI*, vol. 21. Oxford, U.K.:Oxford Univ. Press, 2006, pp. 3 69.
4. D. O. Hebb, *The Organization of Behavior: A Neuropsychological The-*

*ory*. London, U.K.: Psychology Press, 2005.

1. T. Szanda“a, ‘‘Comparison of different learning algorithms for pattern recognition with Hop eld’s neural network,’’ *Procedia Comput. Sci.*, vol. 71, pp. 68 75, Jan. 2015.
2. J. Holmes, ‘‘Knowledge discovery in biomedical data: Theory and methods,’’ in *Methods in Biomedical Informatics*. Cambridge, U.K.: Academic, 2014, pp. 179 240.

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing

1. K. J. Jeffery and I. C. Reid, ‘‘Modi able neuronal connections: An overview for psychiatrists,’’ *Amer. J. Psychiatry*, vol. 154, no. 2,
   1. 156 164, Feb. 1997.
2. E. S. Bromberg-Martin, M. Matsumoto, and O. Hikosaka, ‘‘Dopamine in motivational control: Rewarding, aversive, and alerting,’’ *Neuron*, vol. 68, no. 5, pp. 815 834, Dec. 2010.
3. C. Fyfe, ‘‘Arti cial neural networks and information theory,’’ Dept. Com-put. Inf. Syst., Univ. Paisley, Paisley, U.K., Tech. Rep. Edition 1.2, 2000.
4. R. Rojas, *Neural Networks: A Systematic Introduction*. Berlin, Germany: Springer, 2013.
5. Y. Lou, Z. Ren, Y. Zhao, and Y. Song, ‘‘Using auto-associative neural networks for signal recognition technology on sky screen,’’ in *Proc. Int.* *Conf. Soft Comput. Techn. Eng. Appl.* New Delhi, India: Springer, 2014,
   1. 71 79.
6. R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, ‘‘The capacity of the hop eld associative memory,’’ *IEEE Trans. Inf. Theory*, vol. IT-33, no. 4, pp. 461 482, Jul. 1987.
7. V. Folli, M. Leonetti, and G. Ruocco, ‘‘On the maximum storage capacity of the hop eld model,’’ *Frontiers Comput. Neurosci.*, vol. 10, p. 144, Jan. 2017.
8. D. J. Amit, H. Gutfreund, and H. Sompolinsky, ‘‘Storing in nite numbers of patterns in a spin-glass model of neural networks,’’ *Phys. Rev. Lett.*, vol. 55, no. 14, pp. 1530 1533, Sep. 1985.
9. S. Bartunov, J. W. Rae, S. Osindero, and T. P. Lillicrap, ‘‘Meta-learning deep energy-based memory models,’’ 2019, *arXiv:1910.02720*. [Online]. Available: http://arxiv.org/abs/1910.02720
10. P. Yadav and S. Sharma, ‘‘Enhancing performance of devanagari script recognition using hop eld ANN,’’ *Int. J. Eng. Trends Technol.*, vol. 36, no. 2, pp. 66 75, 2016.
11. P. U. Diehl and M. Cook, ‘‘Unsupervised learning of digit recognition using spike-timing-dependent plasticity,’’ *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.
12. S. Singh, A. Aggarwal, and R. Dhir, ‘‘Use of Gabor lters for recognition of handwritten Gurmukhi character,’’ *Int. J. Adv. Res. Comput. Sci. Softw.* *Eng.*, vol. 2, no. 5, pp. 1 7, 2012.
13. M. Kumar, M. K. Jindal, and R. K. Sharma, ‘‘K-nearest neighbor based of ine handwritten Gurmukhi character recognition,’’ in *Proc. Int. Conf.* *Image Inf. Process.*, Nov. 2011, pp. 1 4.
14. N. Sahu, S. Gupta, and S. Khare, ‘‘Neural network based approach for recognition for Devanagiri characters,’’ *Int. J. Adv. Technol. Eng. Sci.*, vol. 2, no. 5, pp. 187 197, 2014.
15. S. N. Srihari, ‘‘Recognition of handwritten and machine-printed text for postal address interpretation,’’ *Pattern Recognit. Lett.*, vol. 14, no. 4,
    1. 291 302, Apr. 1993.
16. C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, ‘‘A license plate-recognition algorithm for intelligent trans-portation system applications,’’ *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377 392, Sep. 2006.
17. A. Asif, S. A. Hannan, Y. Perwej, and M. A. Vithalrao, ‘‘An overview and applications of optical character recognition,’’ *Int. J. Adv. Res. Sci. Eng*, vol. 3, no. 7, pp. 261 274, 2014.
18. Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, ‘‘An algo-rithm for license plate recognition applied to intelligent transportation system,’’ *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 830 845, Sep. 2011.
19. J. Chang, Y. Zhou, and Z. Liu, ‘‘Limited top-down in uence from recog-nition to same-different matching of chinese characters,’’ *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0156517.
20. R. Parisi, E. D. D. Claudio, G. Lucarelli, and G. Orlandi, ‘‘Car plate recognition by neural networks and image processing,’’ in *Proc. IEEE* *Int. Symp. Circuits Syst. (ISCAS)*, vol. 3, May/Jun. 1998, pp. 195 198.
21. M. Padmanaban and E. Yfantis, ‘‘Handwritten character recognition using conditional probabilities,’’ Univ. Nevada, Las Vegas, NV, USA, Tech. Rep. 1435627.
22. J. A. G. Nijhuis, M. H. Ter Brugge, K. A. Helmholt, J. P. W. Pluim, L. Spaanenburg, R. S. Venema, and M. A. Westenberg, ‘‘Car license plate recognition with neural networks and fuzzy logic,’’ in *Proc. Int. Conf.* *Neural Netw. (ICNN)*, vol. 5, 1995, pp. 2232 2236.
23. P. Wittek, *Quantum Machine Learning: What Quantum Computing*

*Means to Data Mining*. New York, NY, USA: Academic, 2014.

1. R. Chandra, S. Kumar, and P. Goswami, ‘‘Implementation of hop eld neural network for its capacity with nger print images,’’ *Int. J. Comput.* *Appl.*, vol. 141, no. 5, pp. 44 49, 2016.
2. S. Kumar and M. P. Singh, ‘‘Performance evaluation of hop eld neural networks for overlapped english characters by using genetic algorithms,’’ *Int. J. Hybrid Intell. Syst.*, vol. 8, no. 4, pp. 169 184, Nov. 2011.

VOLUME 8, 2020

1. H. Chu, W. Lu, and L. Zhang, ‘‘Application of arti cial neural network in environmental water quality assessment,’’ *J. Agricult. Sci. Technol.*, vol. 15, no. 2, pp. 343 356, 2013.
2. L. Cantini and M. Caselle, ‘‘Hope4Genes: A hop eld-like class predic-tion algorithm for transcriptomic data,’’ *Sci. Rep.*, vol. 9, no. 1, pp. 1 9, Dec. 2019.
3. T. Washizawa, ‘‘Application of hop eld network to saccades,’’ *IEEE* *Trans. Neural Netw.*, vol. 4, no. 6, pp. 995 997, Nov. 1993.
4. N. Soni, E. K. Sharma, and A. Kapoor, ‘‘Application of hop eld neural network for facial image recognition,’’ Blue Eyes Intell. Eng. Sci. Pub., Bhopal, India, Tech. Rep. A2816058119/19 BEIES.
5. N. Soni, N. Singh, A. Kapoor, and E. K. Sharma, ‘‘Low-resolution image recognition using cloud hop eld neural network,’’ in *Progress in* *Advanced Computing and Intelligent Engineering*. Singapore: Springer,2018, pp. 39 46.
6. K. S. Ray and D. Dutta Majumder, ‘‘Application of hop eld neu-ral networks and canonical perspectives to recognize and locate par-tially occluded 3-D objects,’’ *Pattern Recognit. Lett.*, vol. 15, no. 8,
   1. 815 824, Aug. 1994.
7. D. O. Albanez, S. F. da Silva, M. A. Batista, and C. A. Z. Barcelos, ‘‘Images segmentation using a modi ed hop eld arti cial neural net-work,’’ *Proc. Ser. Brazilian Soc. Comput. Appl. Math.*, vol. 6, no. 1,
   1. 1 7, 2018.
8. F. Sabahi, M. O. Ahmad, and M. N. S. Swamy, ‘‘Hop eld network-based image retrieval using re-ranking and voting,’’ in *Proc.* *IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017,
   1. 1 4.
9. X. Yang, L. Zhao, G. M. Megson, and D. J. Evans, ‘‘A system-level fault diagnosis algorithm based on preprocessing and parallel Hop eld neural network,’’ in *Proc. 4th IEEE Workshop RTL High Level Test.*, Nov. 2003,
   1. 189 196.
10. G. Di Modica, S. Gulino, and O. Tomarchio, ‘‘IoT fault management in cloud/fog environments,’’ in *Proc. 9th Int. Conf. Internet Things (IoT)*, 2019, pp. 1 4.
11. H. Yang, B. Wang, Q. Yao, A. Yu, and J. Zhang, ‘‘Ef cient hybrid multi-faults location based on hop eld neural network in 5G coexisting radio and optical wireless networks,’’ *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1218 1228, Dec. 2019.
12. B. Wang, H. Yang, Q. Yao, A. Yu, T. Hong, J. Zhang, M. Kadoch, and M. Cheriet, ‘‘Hop eld neural network-based fault location in wireless and optical networks for smart city IoT,’’ in *Proc. 15th Int. Wireless Commun.* *Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1696 1701.
13. Y. Watanabe, N. Mizuguchi, and Y. Fujii, ‘‘Solving optimization problems by using a hop eld neural network and genetic algo-rithm combination,’’ *Syst. Comput. Jpn.*, vol. 29, no. 10, pp. 68 74, Oct. 1998.
14. R. A. Silver, ‘‘Neuronal arithmetic,’’ *Nature Rev. Neurosci.*, vol. 11, no. 7,
    1. 474 489, Jul. 2010.
15. E. Blem, J. Menon, and K. Sankaralingam, ‘‘Power struggles: Revisiting the RISC vs. CISC debate on contemporary ARM and x86 architectures,’’ in *Proc. IEEE 19th Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2013, pp. 1 12.
16. D. Thain, *Introduction to Compilers and Language Design*. Morrisville, NC, USA: Lulu, 2019.
17. L. Wang, S.-R. Lu, and J. Wen, ‘‘Recent advances on neuromorphic systems using phase-change materials,’’ *Nanosc. Res. Lett.*, vol. 12, no. 1,
    1. 347, Dec. 2017.
18. G. Tanaka, T. Yamane, J. B. HØroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, ‘‘Recent advances in physical reservoir computing: A review,’’ *Neural Netw.*, vol. 115, pp. 100 123, Jul. 2019.
19. V. Saxena, X. Wu, I. Srivastava, and K. Zhu, ‘‘Towards neuromor-phic learning machines using emerging memory devices with brain-like energy ef ciency,’’ *J. Low Power Electron. Appl.*, vol. 8, no. 4, p. 34, 2018.
20. P. Date, C. D. Carothers, J. A. Hendler, and M. Magdon-Ismail, ‘‘Ef - cient classi cation of supercomputer failures using neuromorphic com-puting,’’ in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018,
    1. 242 249.
21. D. Ielmini and S. Ambrogio, ‘‘Emerging neuromorphic devices,’’ *Nan-otechnology*, vol. 31, no. 9, Feb. 2020, Art. no. 092001.
22. S. Furber, ‘‘Large-scale neuromorphic computing systems,’’ *J. Neural* *Eng.*, vol. 13, no. 5, Oct. 2016, Art. no. 051001.
23. K. Roy, A. Jaiswal, and P. Panda, ‘‘Towards spike-based machine intel-ligence with neuromorphic computing,’’ *Nature*, vol. 575, no. 7784,
    1. 607 617, Nov. 2019.

67097

Z. Yu *et al.*: Overview of Neuromorphic Computing

1. X. Yan, Q. Zhao, A. P. Chen, J. Zhao, Z. Zhou, J. Wang, H. Wang, L. Zhang, X. Li, Z. Xiao, K. Wang, C. Qin, G. Wang, Y. Pei, H. Li, D. Ren, J. Chen, and Q. Liu, ‘‘Vacancy-induced synaptic behavior in 2D WS2 nanosheet based memristor for low-power neuromorphic computing,’’ *Small*, vol. 15, no. 24, Jun. 2019, Art. no. 1901423.

ZHEQI YU (Student Member, IEEE) received the bachelor’s degree in electronic information engi-neering, and the master’s degree in information technology from the University of Wolverhamp-ton, in 2014 and 2015, respectively. He is currently pursuing the Ph.D. Research degree with the Uni-versity of Glasgow, Glasgow, U.K. His research interests include low-power neuromorphic sensor fusion, neuromorphic computing, embedded sys-tems, signal processing, medical image analysis, and programmable devices.



AMIR M. ABDULGHANI (Senior Member, IEEE) received the master’s degree from The University of British Columbia, Canada, and the Ph.D. degree from Imperial College London, U.K. He was a Faculty Member with the Department of Electrical and Computer Engineering, Sultan Qaboos University (SQU), Oman, for a period of 18 years. He is currently an Af liate Academic with the University of Glasgow, U.K. He is also the Founder and the Chair of the IEEE Communica-



tion Society, Oman. He was nominated for the Royal Academy of Engineer-ing Research Fellowship, U.K., a research fellowship designed to promote excellence in engineering. He has served as the Acting Assistant Dean for the Training and Community Service, College of Engineering, SQU. He also chaired the Information and Communication Technology (ICT) of the College of Engineering for several years at SQU. He has chaired several other committees with the Department and the College level during his professional career at SQU. His research interests include communication and signals processing, engineering design, entrepreneurship, the Internet of Things (IoT). He is also a member of SQU’s Space Technology Group and the Smart City Research Group established at SQU that is expected to lead the country’s modernization strategy in context of smart city framework. This research group comprises of top national scientists for developing plans and formulating practical smart city solutions. He is actively involved in establishing and nurturing links between academia and industry to encourage and enhance collaboration between the two domains. He has been recognized numerous times for his distinguishing performances by the Minister of Higher Education of Oman, V.C. of SQU, and by several other respected entities including recognition of his services, such as the recently presented Award by the IEEE Oman for his commendable efforts in supporting the section and voluntary services to the IEEE Community.

ADNAN ZAHID (Student Member, IEEE) received the B.Sc. degree (Hons.) in electronics and communications engineering from Glasgow Caledonian University and the M.Sc. degree in electronics and electrical engineering from the University of Strathclyde, in 2016. He is currently pursuing the Ph.D. Research degree with the Uni-versity of Glasgow. His current research interests include machine learning to monitor plant’s health for precision agriculture applications and detection



of water stress in leaves by integrating deep learning and terahertz sensing.

67098

HADI HEIDARI (Senior Member, IEEE) received the Ph.D. degree. He is currently an Assistant Pro-fessor (Lecturer) with the School of Engineering, University of Glasgow, U.K. He leads the Micro-electronics Laboratory and the Research, includes developing microelectronics and sensors for neu-rotechnology devices. He has authored more than 90 articles in tier-1 journals and conferences. He is a member of the IEEE Circuits and Systems Soci-ety Board of Governors (BoG) and the IEEE Sen-



sors Council Administrative Committee (AdCom). He was a recipient of a number of awards, including the best paper awards from ISCAS 2014, PRIME 2014, and ISSCC 2016, the IEEE CASS Scholarship (NGCAS 2017), and the Rewards for Excellence prize from the University of Glasgow, in 2018. He is the General Chair of the 27th IEEE ICECS 2020 in Glasgow. He served on the organizing committee for several conferences including the U.K.-China Emerging Technologies (UCET), PRIME 2015 and 2019, the IEEE SENSORS 2016 and 2017, NGCAS 2017, and BioCAS 2018. He is an Organizer for several special sessions on the IEEE Conferences. As PI, he has received more than £1m funding from major research councils and funding organizations, including the European Commission, EPSRC, the Royal Society, and the Scottish Funding Council. He is involved with the 8.4M EU H2020 FET Proactive on Hybrid Enhanced Regenerative Medicine Systems (HERMES). He is an Associate Editor of the IEEE JOURNAL OF ELECTROMAGNETICS, RF AND MICROWAVES IN MEDICINE AND BIOLOGY,

IEEE ACCESS, *Microelectronics Journal* (Elsevier). He is a Guest Editor of

the IEEE SENSORS JOURNAL.

MUHAMMAD ALI IMRAN (Senior Member, IEEE) received the M.Sc. (Hons.) and Ph.D. degrees from Imperial College London, U.K., in 2002 and 2007, respectively. He is currently the Dean of Glasgow College, UESTC, and a Pro-fessor of communication systems with the School of Engineering, University of Glasgow. He is an Af liate Professor with The University of Okla-homa, USA, and a Visiting Professor with the 5G Innovation Center, University of Surrey, U.K.



He is leading Research with the Scotland 5G Center, University of Glas-gow. He has over 18 years of combined academic and industry experience, working primarily in the research areas of cellular communication systems. He holds over 15 patents. He has authored/coauthored over 400 journal and conference publications. He has been principal/co-principal investigator on over £6 million in sponsored research grants and contracts. He has super-vised 40C successful Ph.D. graduates. He is a Senior Fellow of the Higher Education Academy, U.K. He has an Award of Excellence in recognition of his academic achievements, conferred by the President of Pakistan. He was also awarded the IEEE Comsoc’s Fred Ellersick Award, in 2014, the FEPS Learning and Teaching Award, in 2014, and the Sentinel of Science Award, in 2016. He was twice nominated for the Tony Jean’s Inspirational Teaching Award. He is a shortlisted nalist for The Wharton-QS Stars Awards, in 2014, the QS Stars Reimagine Education Award, in 2016 for innovative teaching, and the VC’s Learning and Teaching Award from the University of Surrey. He is the Editor/Co-Editor of eight books.

VOLUME 8, 2020

Z. Yu *et al.*: Overview of Neuromorphic Computing

QAMMER H. ABBASI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in elec-tronics and telecommunication engineering from the University of Engineering and Technology (UET), Lahore, Pakistan, and the Ph.D. degree in electronic and electrical engineering from the Queen Mary University of London (QMUL), U.K., in 2012. In 2012, he was a Postdoctoral Research Assistant with the Antenna and Electromagnetics Group, QMUL. He is currently a Lecturer (Assis-



tant Professor) with the School of Engineering, University of Glasgow, U.K., and a Researcher Investigator with The Scotland 5G Center. He has con-tributed over 250 leading international technical journal and peer-reviewed

conference papers, and eight books. He received several recognitions for his research, which include appearance on BBC, STV, Dawn News, local and international newspapers, cover of MDPI journal, most downloaded articles, U.K. exceptional talent endorsement by the Royal Academy of Engineering, the National Talent Pool Award with Pakistan, the International Young Scientist Award with NSFC, China, the URSI Young Scientist Award, the National Interest Waiver with USA, four best paper awards, and the Best Representative Image of an Outcome with QNRF. He is an Associate Editor

for the IEEE JOURNAL OF ELECTROMAGNETICS, RF AND MICROWAVES IN MEDICINE AND BIOLOGY, the IEEE SENSORS JOURNAL, the IEEE OPEN ACCESS ANTENNA

AND PROPAGATION, and IEEE ACCESS.He served as a guest editor for numerous special issues in top notch journals.



|  |  |
| --- | --- |
| VOLUME 8, 2020 | 67099 |