IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

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 YU[1], (Student Member, IEEE), AMIR M. ABDULGHANI[1,2], (Senior Member, IEEE), ADNAN ZAHID[1], (Student Member, IEEE), HADI HEIDARI[1], (Senior Member, IEEE), MUHAMMAD ALI IMRAN[1], (Senior Member, IEEE), AND QAMMER H. ABBASI[1], (Senior Member, IEEE)

[1]James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.
[2]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 more unique and novel solutions to the artificial 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 Hopfield algorithms in some large-scale notable hardware projects and seen significant progression. This paper presents a comprehensive review and focuses extensively on the Hopfield 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 artificial intelligence projects.

INDEX TERMS Neuromorphic computing, neuro-inspired model, Hopfield algorithm, artificial intelligence.

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 neuroscientific models and challenging machine learning techniques using. The von Neumann architecture is the computing standard predominantly for machines. However, it has significant differences in organizational structure, power requirements, and processing capabilities relative to the working model of the human brain [1]. Therefore, neuromorphic calculations have emerged in recent years as an auxiliary architecture for the von Neumann system. Neuromorphic 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 defined as neuro-inspired models, algorithms and learning methods, hardware and equipment, support systems and applications [2].

Neuromorphic architectures have several significant and special requirements, such as higher connection and parallelism, low power consumption, memory collocation and processing [3]. Its strong ability to execute complex computational speeds compared to traditional von Neumann architectures, saving power and smaller size of the footprint. These features are the bottleneck of the von Neumann architecture,

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

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

There are ten main motivations for using neuromorphic architecture, including Real-time performance, Parallelism, von Neumann Bottleneck, Scalability, Low power, Footprint, Fault Tolerance, Faster, Online Learning and Neuroscience [1]. Among them, real-time performance is the main driving force of the neuromotor system. Through parallelism 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 development has been low power consumption [5]–[7]. Biological neural networks are fundamentally asynchronous [8], and the brain's efficient data-driven can be based on event-based computational models [9]. However, managing the communication of asynchronous and event-based task in large systems 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 consumption 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 composition of biological neurons is usually composed of cell bodies, axon, and dendrites. The neuron models implemented by each component of the specified model are divided into five groups, based on the type of model being distinguished by biologically and computationally driven.

(1) Biologically Plausible [12]: Specifically 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 equation [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 simplified biologically plausible 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 randomness [16].

(2) Biologically-Inspired [17]: Ignore biological rationality 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 dynamics of pulse firing, cluster firing, and chaotic behavior of neurons [19].

(3) Neuron and other Biologically- Inspired Mechanism [20]: Neuron models include other biologically inspired components.

(4) Integrate and-Fire Neurons [21]: Biology-inspired spike neuron model.

(5) 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 categories. One of the synaptic models is bio-inspired synaptic implementations that include spike-based systems and feed-forward neural networks [23]. For more complex synaptic 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 neuromorphic 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], artificial neural network cellular neural networks [27] and pulse-coupled neural networks [28], cellular automata [29], fuzzy neural networks [30]. Hopfield 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 classification [33], etc.

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

Neuromorphic implementation based on high-level standards is to divide hardware implementation into three categories: digital, analog, and hybrid analog/digital platforms [17]. The analog system takes advantage of the physical 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 physical 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 first one for FPGA, which has been used frequently in neuromorphic systems [17]. Another one is custom or application-specific integrated circuit (ASIC) chips, which is also common neuromorphic implementations [35].

For neuromorphic systems, custom analog integrated circuits have several universal features which make them suitable for each other. There are all properties that occur in both analog circuits and biological systems, such as the conservation of charge, amplification, thresholding and integration [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 systems about unreliability can be addressed by using digital components. Meanwhile, analog neuromorphic systems of synaptic weights are often stored in digital memories. Neuromorphic 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 system [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 neuromotor system that is based on a software-based simulator 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, including images [40], speech [41], data classification [42], control [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 Neumann architecture.

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

II. HOPFIELD ALGORITHM

The Hopfield network is an important algorithm in the history of neural network development. Hopfield [45], a physicist at the California Institute of Technology, proposed in 1982 that it is a single-layer feedback neural network. Hopfield neural network is the simplest and most applicable model in feedback 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 contaminated by noise.

The Hopfield 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 flux. The feedback network can be divided as stable and unstable, which is by judging its stability. For a Hopfield network, the key is to determine its weight coefficient under stable conditions [48]. If the weight matrix W of the Hopfield 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 Hopfield network is divided into two types: discrete Hopfield neural network (DHNN) and continuous Hopfield neural network (CHNN) [49]. Discrete Hopfield Neural Network (DHNN): The output of a neuron takes only 1 and 0, which is respectively indicating the neuron in an activation and inhibition state [50]. Continuous Hopfield 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 Hopfield network, the output data is equal to the mode size and dimension 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 W_{ij} [53]. So for a Hopfield 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 Hopfield network is a feature that can be used for associative memory. This is one of the intelligent characteristics of human beings, so the Hopfield algorithm can simulate human ‘‘want’’ [55]. By reviewing and thinking about the past scenes, it is used as the associative memory of the Hopfield Neural network. Firstly, learning training process to determine the weight coefficient 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 coefficient 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 Hopfield neural network concept is designed closely related to circuit hardware deployment [58]. The main idea of the Hopfield is the use of the hardware circuit to simulate neural network optimization process. This process can be fast that takes an analog circuit processing advantage rather than digital circuit [59]. Unlike the software realization of the Hopfield neural network, the hardware implementation of the algorithm makes brain-like computations possible [60].

Hopfield 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 clarified the relationship between the neural network and dynamics model [61]. Then, established the stability criterion 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 Hopfield network. In addition, Hopfield algorithm compares the feedback network with the Ising model in statistical physics and defines 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 field of neural networks. In 1984, Hopfield designed and developed the circuit of the network algorithm model [63], it is stating that neurons can be implemented with operational amplifiers, and all neuron connections can be simulated by electronic circuits [64]. One of the continuous Hopfield networks using circuit deployed, which is successfully solved travelling salesman problem (TSP) calculation problem. It proves that the Hopfield circuit can address the optimization problem [65].

Moreover, Hopfield network can convert analog signals into a digital format that is to realise associative memory, signal estimation and combination optimization applications [66]. This solution is similar to the method of the human first layer to achieve signal processing. So, it belongs to the neuromorphic calculation. Due to the algorithm stability of output digital signal, Hopfield 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

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

Discrete Hopfield network of function that simulates the memory of biological neural network is often called associative memory network. Associative memory (AM) is an integral part of neural network theory [71], and it is a significant function in artificial intelligence and other fields 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 tolerance of neural networks to restore incomplete, defaced and distorted input samples achieve complete prototypes, which are suitable for recognition, classification purposes [76].

Association is based on memory where the information is stored first, and then retrieved in a certain way or rule. Associative memory is also called content-addressed memory, which means the process of associative memory is the process of accessing information [77]. Information is distributed in the content of biological memory, rather than a specific address. The storage of information is distributed, not centralized. The storage of information is content addressable 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, Hopfield 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 commonly used associative memory [80]. The common dynamic memory network is the Hopfield model (auto-associative memory) [81] and Kosko's Bidirectional Associative Memory (BAM) model (hetero-associative memory) [82].

The applied classification based on associative memory 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 Hopfield 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 final steady state as attractor dynamics after the weights are adjusted, it is according to the given incomplete or affected information as the associative keyword [85]. In fact, this associative memory process is the continuous movement of the energy function inside the Hopfield 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, Hopfield algorithm is a powerful learning system with simple structure and easy programming. The Hopfield 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

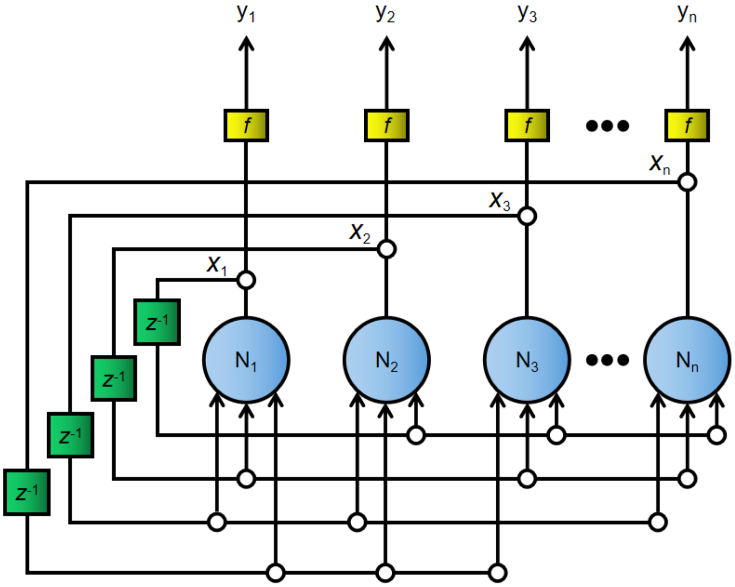

FIGURE 1. The fully connected network architecture of the Hopfield network (adapted from [91]).

state, which is the output results. Therefore, the state of the Hopfield network evolves in the direction of decreasing energy function. Since the energy function is bounded, the system will incline to a stable state, which is the output of the Hopfield network [87].

From a system perspective, the feedforward neural network model has limited computing power, and the feedback dynamics of a feedback neural network more stronger computing power than a feedforward neural network, which is based on feedback to enhance global stability [88]. In feedback 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 application is properly set, Hopfield Neural networks can be more robust on the applications. It is a static nonlinear mapping, and the nonlinear processing capability of the complex system can be obtained by the compound mapping of the simple nonlinear processing unit [90].

The discrete Hopfield neural network (DHNN) which is also known as a feedback neural network is fully connected 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 z^{-1} [91]. In the Hopfield 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 as:

$$x_i(t) = \sum_{j=1}^{n} w_{ij} y_j(t) \tag{1}$$

In the time $t+1$, the output of y in the neuron i can be expressed as:

$$y_i(t+1) = f(x_i(t)) \tag{2}$$

where $f(\cdot)$ is the transfer function:

$$f(a) = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases} \tag{3}$$

Converting the network structure into a circuit topology that is shown in Fig 2, Hopfield neural networks are equivalent to amplified 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 amplifiers. It includes a non-inverting amplifier and an inverting amplifier (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 I_s ($S = 1,2,3,\ldots, N$) [94]. At the connection point of each neuron, there is has a resistor, which represents the impedance $\mathrm{R}_{ij}(\mathrm{R}_{ij} = 1/W_{ij})$ connected to other neurons [95], and the constant W_{ij} represents the network weight between neuron i and neuron j.

For the bias current calculation is shown in the following:

$$I_i = \sum_{j=1}^{s} \frac{x_i}{r_{ij}} = \sum_{j=1}^{s} x_j w_{ij} \tag{4}$$

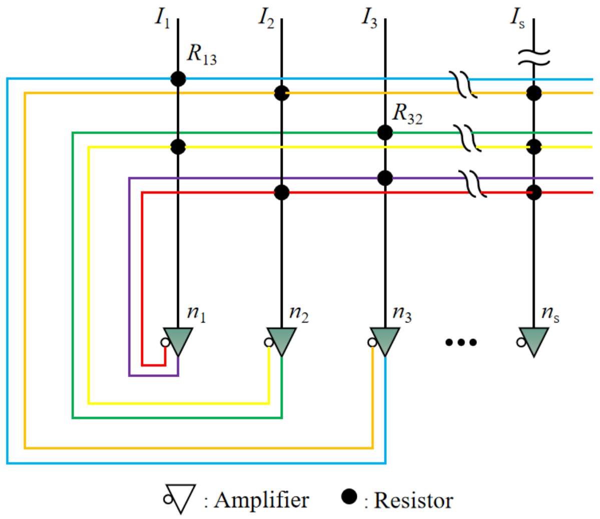

FIGURE 2. Circuit topology of Hopfield network.

IV. LEARNING METHOD

The neural network learning method can be classified 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 Hopfield algorithm training is based on the dynamics evolving in discrete time with time steps to discrete learning [99].

According to different learning environments, neural network learning methods can be divided into supervised learning 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 training [102]. If the sample situation is changed, then weights can be modified to adapt to the new environment after training [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 classic example of unsupervised learning is the Hebb learning rule [108].

Hebb learning rules are the basis of artificial neural networks. The adaptability of neural networks is realised through 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 artificial 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 models. 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 reflected 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 relationship 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 field, when the A cell's neuron axon is close enough to B cells, it repeatedly 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 follows:

$$W_{ij}(t+1) = W_{ij}(t) + a * y_i * y_j \tag{5}$$

The W_{ij} represents the connection weight of neuron j to neuron i, y_i and y_j represent the output of two neurons, 'a' is a constant representing the learning rate. If y_i and y_j are activated at the same time, that means y_i and y_j are positive, and the W_{ij} will increase. If y_i is activated and y_j is inhibited, that means y_i is positive and y_j is negative, then W_{ij} will decrease. This equation shows that the change in weight W_{ij} 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].

Hopfield 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, $P = 1, 2, 3, \ldots, P$, $x \in \{-1, 1\}^n$, and the samples are orthogonal to each other. The $n> p$, then the weight matrix is outer product sum of memory samples [117].

Outer product sum:

$$W = \sum_{P=1}^{P} x^p \left(x^p\right)^T \tag{6}$$

Component-wise manner:

$$W_{ij} = \begin{cases} \sum_{P=1}^{P} x_i^p x_j^p, & i \neq j \\ 0, & i = j \end{cases} \tag{7}$$

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

Since P samples X^p, $P = 1, 2, 3, \ldots, P$, $x \in \{-1, 1\}^n$ is pairwise orthogonal. The calculation according to the following equation:

$$\left(x^p\right)^T x^k = \begin{cases} 0 & p \neq k \\ n & p = k \end{cases} \tag{8}$$

Due to $n > p$, the attractor x^p will be calculated by the following

$$f\left(wx^p\right) = f\left[(\text{n-p})\, x^p\right] = sgn\left[(\text{n-p})\, x^p\right] = x^p \tag{9}$$

The weights computing workflow of Hopfield network [118] as following:

```
Algorithm 1 Hopfield Algorithm Workflow
Input: $P_n$ pattern samples ($n$ dimensional orthogonal vector);
1: Set to the initial state of the network X = P;
2: Set the number of iteration steps;
3: Calculate the $W$ weight of the network: $W = \sum_{i=1}{}^{n}\left[P^T P - I\right]$;
4: Since $W_{ij} = 0$, subtract the unit matrix I;
5: Perform iterative calculation;
6: 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 Hopfield Neural Network
```

However, when the network size is fixed, 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 network 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]. Nonetheless, 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 rate, which needs the smaller information storage capacity of the network. When it exceeds $0.14n$ 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].

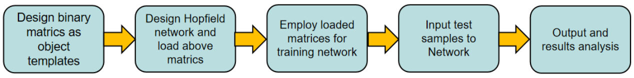
Design binary matrics as object templates
Design Hopfield network and load above matrics
Employ loaded matrices for training network
Input test samples to Network
Output and results analysis

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

V. APPLICATIONS

In daily life, character recognition has high practical application value in the postal [128], transportation [129] and document management process [130], such as the recognition 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 character cannot be obtain and identification of noisy characters become a key issue [132].

At present, there are several methods for character recognition, which are mainly divided into a neural network [133], probability statistical [134], and fuzzy recognition [135]. The traditional character recognition method cannot recognize well under the condition of noise interference. However, the discrete Hopfield 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 discrete Hopfield neural network concept, and the network can recognize the 10 digits that fall in the range from 0-9. Furthermore, despite any disturbance by certain noise due to the specified 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 defined -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 Hopfield network of recognition application workflow is depicted as below:

TABLE 1. Algorithm comparison on the character recognition cases.

Algorithm	Pre-Processing	Training Dataset Sizes	Accuracy
Hopfield	Gabor Filter, Freeman Chain Code	167	96% [123]
SNN(Spiking Neural Networks)	Poisson-spike	60000	95% [124]
SVM (Support Vector Machine)	Gabor filter	700	94.2% [125]
KNN (K-Nearest Neighbour)	Diagonal feature extraction	3500	94.12% [126]
Back Propagation	Gradient Feature	210	91.2% [127]

1) STANDARD SAMPLES

In this study, standard samples are selected to convert into binarisation matrix. In this application of characters recognition, 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 Hopfield network should contain 100 neurons, 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 finally 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 $X(t + 1) = X(t)$.

When the X state is reached to a steady-state, then it is considered as an attractor. For the attractors, it is determined of final behaviour on a dynamic system. The system requirement 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.

4) 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 Hopfield 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.

5) RESULT ANALYSIS

The test results are analyzed and compared with the trained object vector. Through a confidence level calculation, the resulting output the closest of the object vector can be obtained. Finally, the classification result is achieved. Further research shows the recognition effect decreases by the increase of noise intensity. When the noise intensity exceeds 30%, the Hopfield 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 Hopfield algorithm. And compared accuracy with in the character recognition application. The Hopfield algorithm is most dataset request and best accuracy on the application, which is means the Hopfield algorithm is suitable on this character recognition field.

B. OTHER APPLICATIONS

Based on the discrete Hopfield neural network, it has the function of associative memory. In recent years, many researchers have attempted to apply Hopfield network to various fields in order to replace conventional techniques to address the issues, such as water quality evaluation [139] and generator fault diagnosis [147], and have achieved considerable results by applying aforementioned method. For example, in the Internet of Things (IoT) applications, where multiple 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 Hopfield Neural Network [149]. The built-in Hopfield 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 applicability for the field. When the Hopfield neural network collaborates with some notable optimisation algorithms, not the network alone, that provides its associative memory stronger but also improves the application efficiency. For example, the existence of many pseudo-stable points in general discrete Hopfield neural networks, limit the proficiency of network. Therefore, a Genetic algorithm can be considered for the discrete Hopfield network [151], and the global search capability of the Genetic algorithm is used to optimise the steady-state of Hopfield associative memory function. So that makes the associative mode jump out of the pseudo-stable point, and then the Hopfield network maintains a high associative success rate under the condition of higher noise to signal ratio.

C. APPLICATION COMPARISON

The workflow details of Hopfield algorithm are similar to those of above in section B mentioned. However, the main

TABLE 2. Algorithm extraction and use cases for Hopfield.

Paper	Task	Data	Method	Problem Address
[139]	Data Classification	Seven parameters were selected from the 12 original parameters by Factor analysis. (*BOD5, CODMN, NH3-N, Cu, Zn, Pb, TN*)	Use positive and negative values (+1, -1) to represent whether water quality monitoring under different parameters is passed, which to set different memory patterns for water quality levels corresponding to the network.	Water quality evaluation
[140]		The network is constructed with different genes signatures as nodes, class 1 markers are shown in red, and class 2 markers are shown in blue to store into the model.	First, the expression value of the signature gene in the sample is discretized to +1, -1, after that the value is assigned to the corresponding node. Then, when all values of only one classe are +1 that to evolve the model. Finally, the sample is assigned to the convergent class.	Classify to transcriptomic data of Gene
[141]	Pattern Recognition	Four templates (8 x 8 binary valued pixels)	This network designs three types of neuron blocks, called saccade (S) block, hidden (H) block and output (O) block. The S block is a matrix representation of the position of the input mode neurons. O block of neurons represent the class of the input pattern. H block is a template for the target pattern class.	Speed up saccadic tasks
[140]		Eye's iris information into 150 × 150 pixels image input to network	The input image is first scaled to a 20x20 binarisation matrix, and then multiplied with the 400 x 400 value matrix. The input value of weight is obtained by performing a dot product calculation to the input vector.	Recognition of eye's iris
[142] [143]	Image Recognition	Convert the face image into a binarized matrix and store it in weights using a Hebb rule	Reduce the image to 60 x 60 size, and then transformation to 1D row vector store into matrix. And calculate the similarity as result by measured network output and stable image.	Retrieval of distorted face images
[144]	Feature Match	Representative edge-point that establish by scene edge and the edge of the model object	Feature matching is performed of a two-dimensional array. The row data means the features of the scene, and the columns means the features of the object model. The neuron output the similarity, which compared to the two features the scene and the object model.	Recognise and locate partially occluded 3D rigid objects
[145]	Image segmentation	A random assignment of N pixels to M classes	Construct a network model of N × M neurons, it is the columns means cluster and the rows means pixel, which to classify the feature space.	Images segmentation for satellite images
[87]		Mammograms image of feature vector that containing gray value and edge information, input each discrete pixel position into N x N matrix.	The network deployed a binarized N × M neurons matrix to achieve image segmentation in the image. N is the number of sampling points of the initial contour, and M is the number of grid points on each grid line.	Tumor Boundary Detection
[146]	Image retrieval	Subdividing image into L blocks, and extracting features that to generate feature vectors input to network	According to position to calculate weight matrix for each block of all images. Use the method of bootstrapping to get the final weight parameters.	Identify similar images by visual contents

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

The actual practice of Hopfield neural network involves a surprising number of scientific disciplines, which is the key technical areas it covers include Data Classification, Pattern Recognition, Image Recognition, Feature Match, Image segmentation, Image retrieval. The result of the investigation was reported as shown in Table 2. By comparing the applications in various fields, we can find 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], the conventional processor instruction set (including × 86 and ARM, etc.) was developed for general-purpose computing [153]. These operations are arithmetic operations (addition, subtraction, multiplication and division) and logical operations (AND-OR-NOT) [154]. It often requires hundreds or thousands of instructions to complete the processing of neuron computing, making the low processing efficiency of the hardware inefficient.

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 computing 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 efficiency 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 computing design will be used as the underlying architecture to improve the neural network computing in the long-term planning [156].

Among the above, ASIC is a kind of chip specifically designed for this special purpose. Compared to FPGA, it features stronger performance, smaller size, less power consumption, lower cost and more progress in developing hardware 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 confidentiality, high computing performance and high computing efficiency 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 transistors [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 artificial synapses and excitement need improvement [160]. Nowadays, there are a lot of explorations on how to simulate or create synthetic synapses. Taken as a whole, for produced formal neuronal 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 efficiency is low, causing neuronal simulation efficiency 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 plastic 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 hardware [162].

Eventually, the architecture, algorithm and programming scheme of adaptive neuromorphic computing is in a wide blank and a long way to reach a final goal that replaces to von Neumann's structure in the artificial 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 limited applications. In addition, there are many software-based neural network applications that has been deployed, but hardware-based neural network design has been the key to the neuromorphic design. Convention neural network circuit implementation is thought of time-consuming and inconvenient. In order to apply a simple and fast design method to neural network hardware, which can optimise and manufacture neuromorphic computing systems, needs to systematically unificate the requirements of the software calculation process. Furthermore, it can process and improves the final software-level application indicators to quantify hardware attributes. Finally,a testable solution for a specification component can be achieved.

In this study, we have attempted to give an overview of work that has been carried out in the hardware implementation field on neural networks. In addition, we have also discussed the various techniques and methods employed in the overall progression and implementation of Hopefield Algorithm. In this regard, it is found that this algorithm has been extensively deployed in various disciplines based on feasibility and efficiency. Moreover, we have also highlighted 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 significant 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 significant step towards the hardware implementation of low-power neuromorphic processing systems using advanced Hopfield 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, G. 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, "Neuromorphic computing—From materials research to systems architecture roundtable," USDOE Office 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 E. 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.

[10] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, Feb. 2018.

[11] G. Indiveri and S.-C. Liu, "Memory and information processing in neuromorphic systems," *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, Aug. 2015.

[12] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J. Physiol.*, vol. 117, no. 4, pp. 500–544, 1952.

[13] Q. Ma, M. R. Haider, V. L. Shrestha, and Y. Massoud, "Bursting Hodgkin–Huxley model-based ultra-low-power neuromimetic silicon neuron," *Anal. Integr. Circuits Signal Process.*, vol. 73, no. 1, pp. 329–337, Oct. 2012.

[14] A. Borisyuk, *Morris-Lecar Model*. New York, NY, USA: Springer, 2015, pp. 1758–1764, doi: 10.1007/978-1-4614-7320-6_150-1.

[15] M. Graupner and N. Brunel, "Calcium-based plasticity model explains sensitivity of synaptic changes to spike pattern, rate, and dendritic location," *Proc. Nat. Acad. Sci. USA*, vol. 109, no. 10, pp. 3991–3996, Mar. 2012.

[16] A. Galves and E. Löcherbach, "Infinite 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.

[17] F. Grassia, T. Levi, T. Kohno, and S. Saïghi, "Silicon neuron: Digital hardware implementation of the quartic model," *Artif. Life Robot.*, vol. 19, no. 3, pp. 215–219, Nov. 2014.

[18] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Trans. Neural Netw.*, vol. 15, no. 5, pp. 1063–1070, Sep. 2004.

[19] 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.

[20] J. V. Arthur and K. Boahen, "Silicon neurons that inhibit to synchronize," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, p. 4.

[21] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[22] 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.

[23] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, "Deep learning in spiking neural networks," *Neural Netw.*, vol. 111, pp. 47–63, Mar. 2019.

[24] J. H. Kotaleski and K. T. Blackwell, "Modelling the molecular mechanisms of synaptic plasticity using systems biology approaches," *Nature Rev. Neurosci.*, vol. 11, no. 4, pp. 239–251, Apr. 2010.

[25] E. Neftci, "Stochastic neuromorphic learning machines for weakly labeled data," in *Proc. IEEE 34th Int. Conf. Comput. Design (ICCD)*, Oct. 2016, pp. 670–673.

[26] E. Donati, F. Corradi, C. Stefanini, and G. Indiveri, "A spiking implementation of the lamprey's central pattern generator in neuromorphic VLSI," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, pp. 512–515.

[27] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 10, pp. 1273–1290, Oct. 1988.

[28] 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.

[29] 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.

[30] 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.

[31] Z. Uykan, "Continuous-time hopfield neural network-based optimized solution to 2-channel allocation problem," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 23, no. 2, pp. 480–490, 2015.

[32] R. T. Cabeza, E. B. Viciedo, A. Prieto-Moreno, and V. M. Vega, "Fault diagnosis with missing data based on hopfield neural networks," in *Mathematical Modeling and Computational Intelligence in Engineering Applications*. Cham, Switzerland: Springer, 2016, pp. 37–46.

[33] X. Hu and T. Wang, "Training the hopfield neural network for classification using a STDP-like rule," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 737–744.

[34] G. Indiveri, "Computation in neuromorphic analog VLSI systems," in *Neural Nets WIRN Vietri-01*. London, U.K.: Springer, 2002, pp. 3–20.

[35] F. Blayo and P. Hurat, "A VLSI systolic array dedicated to hopfield neural network," in *VLSI for Artificial Intelligence*. Boston, MA, USA: Springer, 1989, pp. 255–264.

[36] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629–1636, Oct. 1990.

[37] J. V. Arthur and K. Boahen, "Learning in silicon: Timing is everything," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 75–82.

[38] İ. Bayraktaroğlu, A. S. Öğrenci, G. Dündar, S. Balkır, and E. Alpaydın, "ANNSyS: An analog neural network synthesis system," *Neural Netw.*, vol. 12, no. 2, pp. 325–338, Mar. 1999.

[39] D. Brüderle *et al.*, "A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems," *Biol. Cybern.*, vol. 104, nos. 4–5, pp. 263–296, May 2011.

[40] 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.

[41] 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.

[42] 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.

[43] 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 first step towards neuromorphic applications," *Phys. Status Solidi A*, vol. 212, no. 2, pp. 239–244, Feb. 2015.

[44] Y. Chen, G. Liu, C. Wang, W. Zhang, R.-W. Li, and L. Wang, "Polymer memristor for information storage and neuromorphic applications," *Mater. Horizons*, vol. 1, no. 5, pp. 489–506, 2014.

[45] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.

[46] X. S. Zhang, *Neural Networks in Optimization*, vol. 46. Boston, MA, USA: Springer, 2013.

[47] 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 reconfigurable memristive hopfield neural network," *Nature Commun.*, vol. 6, no. 1, pp. 1–8, Nov. 2015.

[48] J. Schmidhuber, "Habilitation thesis: System modeling and optimization," Page 150 ff Demonstrates Credit Assignment Across Equivalent 1,200 Layers Unfolded RNN, Tech. Rep. 15-04-1993, 1993.

[49] R. Ma, Y. Xie, S. Zhang, and W. Liu, "Convergence of discrete delayed hopfield neural networks," *Comput. Math. Appl.*, vol. 57, nos. 11–12, pp. 1869–1876, Jun. 2009.

[50] C. Ramya, G. Kavitha, and D. K. S. Shreedhara, "Recalling of images using hopfield neural network model," 2011, *arXiv:1105.0332*. [Online]. Available: http://arxiv.org/abs/1105.0332

[51] A. A. Adly and S. K. Abd-El-Hafiz, "Efficient vector hysteresis modeling using rotationally coupled step functions," *Phys. B, Condens. Matter*, vol. 407, no. 9, pp. 1350–1353, May 2012.

[52] U.-P. Wen, K.-M. Lan, and H.-S. Shih, "A review of hopfield neural networks for solving mathematical programming problems," *Eur. J. Oper. Res.*, vol. 198, no. 3, pp. 675–687, Nov. 2009.

[53] S. Kumar and M. P. Singh, "Pattern recall analysis of the hopfield neural network with a genetic algorithm," *Comput. Math. Appl.*, vol. 60, no. 4, pp. 1049–1057, Aug. 2010.

[54] J. D. Keeler, "Comparison between Kanerva's SDM and hopfield-type neural networks," *Cognit. Sci.*, vol. 12, no. 3, pp. 299–329, Jul. 1988.

[55] M. A. Akra, "On the analysis of the Hopfield network: A geometric approach," Ph.D. dissertation, Massachusetts Inst. Technol., Cambridge, MA, USA, 1988.

[56] B. Müller, J. Reinhardt, and M. T. Strickland, *Neural Networks: An Introduction*. Berlin, Germany: Springer, 2012.

[57] J. A. Anderson, *An Introduction to Neural Networks*. Cambridge, MA, USA: MIT Press, 1995.

[58] H. K. Sulehria and Y. Zhang, "Study on the capacity of hopfield neural networks," *Inf. Technol. J.*, vol. 7, no. 4, pp. 684–688, 2008.

[59] 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.

[60] 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.

[61] 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.

[62] S. Cocco, R. Monasson, L. Posani, S. Rosay, and J. Tubiana, "Statistical physics and representations in real and artificial neural networks," *Phys. A, Stat. Mech. Appl.*, vol. 504, pp. 45–76, Aug. 2018.

[63] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: A model," *Science*, vol. 233, no. 4764, pp. 625–633, Aug. 1986.

[64] D. Tank and J. Hopfield, "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.

[65] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, no. 3, pp. 141–152, 1985.

[66] A. Tankimanova, A. K. Maan, and A. P. James, "Level-shifted neural encoded analog-to-digital converter," in *Proc. 24th IEEE Int. Conf. Electron., Circuits Syst. (ICECS)*, Dec. 2017, pp. 377–380.

[67] M. S. Badie and O. K. Ersoy, "A multistage approach to the Hopfield model for bi-level image restoration," Purdue Univ., West Lafayette, IN, USA, ECE Tech. Rep. TR-ECE 95-22, 1995, Paper 143.

[68] S. Draghici, "Neural networks in analog hardware—Design and implementation issues," *Int. J. Neural Syst.*, vol. 10, no. 1, pp. 19–42, 2000.

[69] Y. Zhu and Z. Yan, "Computerized tumor boundary detection using a hopfield neural network," *IEEE Trans. Med. Imag.*, vol. 16, no. 1, pp. 55–67, Feb. 1997.

[70] D.-Q. Wang and Z.-J. Li, "Pattern recognition based on hopfield neural network," *J. Wuhan Yejin Univ. Sci. Technol.*, vol. 4, pp. 231–236, Apr. 2005.

[71] G. Palm, "Neural associative memories and sparse coding," *Neural Netw.*, vol. 37, pp. 165–171, Jan. 2013.

[72] J. K. Paik and A. K. Katsaggelos, "Image restoration using a modified hopfield network," *IEEE Trans. Image Process.*, vol. 1, no. 1, pp. 49–63, Jan. 1992.

[73] B. Bavarian, "Introduction to neural networks for intelligent control," *IEEE Control Syst. Mag.*, vol. 8, no. 2, pp. 3–7, Apr. 1988.

[74] S. Matsuda, "'Optimal' Hopfield network for combinatorial optimization with linear cost function," *IEEE Trans. Neural Netw.*, vol. 9, no. 6, pp. 1319–1330, Nov. 1998.

[75] J.-S. Jang, S.-W. Jung, S.-Y. Lee, and S.-Y. Shin, "Optical implementation of the Hopfield model for two-dimensional associative memory," *Opt. Lett.*, vol. 13, no. 3, pp. 248–250, Mar. 1988.

[76] 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 Hopfield Neural Network on FPGAs," in *Neurocomputing*, vol. 171. Amsterdam, The Netherlands: Elsevier, 2016, pp. 1606–1609.

[77] S. S. Yau and H. Fung, "Associative processor architecture—A survey," *ACM Comput. Surv.*, vol. 9, no. 1, pp. 3–27, 1977.

[78] 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.

[79] L. D. Pyeatt, *Modern Assembly Language Programming With the ARM Processor*. London, U.K.: Newnes, 2016.

[80] M. Gupta, L. Jin, and N. Homma, *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory*. Hoboken, NJ, USA: Wiley, 2004.

[81] Y. P. Singh, A. Khare, and A. Gupta, "Analysis of Hopfield autoassociative memory in the character recognition," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 3, pp. 500–503, 2010.

[82] X. Zhuang, Y. Huang, and S.-S. Chen, "Better learning for bidirectional associative memory," *Neural Netw.*, vol. 6, no. 8, pp. 1131–1146, Jan. 1993.

[83] 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.

[84] B. Ripley, "Linear discriminant analysis," in *Pattern Recognition and Neural Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2007, pp. 91–120.

[85] 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.

[86] F. A. Unal, "Temporal pattern matching using an artificial neural networs," in *Neural Networks and Pattern Recognition*. Amsterdam, The Netherlands: Elsevier, 1998, pp. 77–104.

[87] A. Meyer-Baese and V. J. Schmid, *Pattern Recognition and Signal Analysis in Medical Imaging*. Amsterdam, The Netherlands: Elsevier, 2014.

[88] F. Mastrogiuseppe and S. Ostojic, "A geometrical analysis of global stability in trained feedback networks," *Neural Comput.*, vol. 31, no. 6, pp. 1139–1182, Jun. 2019.

[89] J. F. Pagel and P. Kirshtein, *Machine Dreaming and Consciousness*. New York, NY, USA: Academic, 2017.

[90] G. Serpen, "Hopfield network as static optimizer: Learning the weights and eliminating the guesswork," *Neural Process. Lett.*, vol. 27, no. 1, pp. 1–15, Feb. 2008.

[91] L. Rong and Q. Junfei, "A new water quality evaluation model based on simplified hopfield neural network," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 3530–3535.

[92] 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.

[93] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. USA*, vol. 81, no. 10, pp. 3088–3092, May 1984.

[94] C. V. Negoita, *Cybernetics and Applied Systems*. Boca Raton, FL, USA: CRC Press, 1992.

[95] 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.

[96] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Joint modeling of event sequence and time series with attentional twin recurrent neural networks," 2017, *arXiv:1703.08524*. [Online]. Available: http://arxiv.org/abs/1703.08524

[97] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*. Hoboken, NJ, USA: Wiley, 2015.

[98] S. Roa and F. Nino, "Classification of natural language sentences using neural networks," in *Proc. Flairs Conf.*, 2003, pp. 444–449.

[99] P. Zheng, J. Zhang, and W. Tang, "Analysis and design of asymmetric hopfield networks with discrete-time dynamics," *Biol. Cybern.*, vol. 103, no. 1, pp. 79–85, Jul. 2010.

[100] J. Torres, Ed., *First Contact With Deep Learning: Practical introduction With Keras* (WATCH THIS SPACE collection—Barcelona Book 5). Barcelona, Spain: Independently Published, 2018.

[101] R. Reed and R. J. MarksII, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1999.

[102] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," 2017, *arXiv:1710.02913*. [Online]. Available: http://arxiv.org/abs/1710.02913

[103] 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.

[104] X. Wu, J. Ghaboussi, and J. H. Garrett, "Use of neural networks in detection of structural damage," *Comput. Struct.*, vol. 42, no. 4, pp. 649–659, Feb. 1992.

[105] 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.

[106] S. Marinai and H. Fujisawa, *Machine Learning in Document Analysis and Recognition*, vol. 90. Berlin, Germany: Springer, 2007.

[107] G. Zhang, *Star Identification: Methods, Techniques and Algorithms*. Berlin, Germany: Springer, 2019.

[108] 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.

[109] 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

[110] 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.

[111] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. London, U.K.: Psychology Press, 2005.

[112] T. Szandała, "Comparison of different learning algorithms for pattern recognition with Hopfield's neural network," *Procedia Comput. Sci.*, vol. 71, pp. 68–75, Jan. 2015.

[113] J. Holmes, "Knowledge discovery in biomedical data: Theory and methods," in *Methods in Biomedical Informatics*. Cambridge, U.K.: Academic, 2014, pp. 179–240.

[114] K. J. Jeffery and I. C. Reid, "Modifiable neuronal connections: An overview for psychiatrists," *Amer. J. Psychiatry*, vol. 154, no. 2, pp. 156–164, Feb. 1997.

[115] 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.

[116] C. Fyfe, "Artificial neural networks and information theory," Dept. Comput. Inf. Syst., Univ. Paisley, Paisley, U.K., Tech. Rep. Edition 1.2, 2000.

[117] R. Rojas, *Neural Networks: A Systematic Introduction*. Berlin, Germany: Springer, 2013.

[118] 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, pp. 71–79.

[119] R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh, "The capacity of the hopfield associative memory," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 4, pp. 461–482, Jul. 1987.

[120] V. Folli, M. Leonetti, and G. Ruocco, "On the maximum storage capacity of the hopfield model," *Frontiers Comput. Neurosci.*, vol. 10, p. 144, Jan. 2017.

[121] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing infinite numbers of patterns in a spin-glass model of neural networks," *Phys. Rev. Lett.*, vol. 55, no. 14, pp. 1530–1533, Sep. 1985.

[122] 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

[123] P. Yadav and S. Sharma, "Enhancing performance of devanagari script recognition using hopfield ANN," *Int. J. Eng. Trends Technol.*, vol. 36, no. 2, pp. 66–75, 2016.

[124] 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.

[125] S. Singh, A. Aggarwal, and R. Dhir, "Use of Gabor filters for recognition of handwritten Gurmukhi character," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 5, pp. 1–7, 2012.

[126] M. Kumar, M. K. Jindal, and R. K. Sharma, "K-nearest neighbor based offline handwritten Gurmukhi character recognition," in *Proc. Int. Conf. Image Inf. Process.*, Nov. 2011, pp. 1–4.

[127] 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.

[128] S. N. Srihari, "Recognition of handwritten and machine-printed text for postal address interpretation," *Pattern Recognit. Lett.*, vol. 14, no. 4, pp. 291–302, Apr. 1993.

[129] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377–392, Sep. 2006.

[130] 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.

[131] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 830–845, Sep. 2011.

[132] J. Chang, Y. Zhou, and Z. Liu, "Limited top-down influence from recognition to same-different matching of chinese characters," *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0156517.

[133] 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.

[134] M. Padmanaban and E. Yfantis, "Handwritten character recognition using conditional probabilities," Univ. Nevada, Las Vegas, NV, USA, Tech. Rep. 1435627.

[135] 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.

[136] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. New York, NY, USA: Academic, 2014.

[137] R. Chandra, S. Kumar, and P. Goswami, "Implementation of hopfield neural network for its capacity with finger print images," *Int. J. Comput. Appl.*, vol. 141, no. 5, pp. 44–49, 2016.

[138] S. Kumar and M. P. Singh, "Performance evaluation of hopfield neural networks for overlapped english characters by using genetic algorithms," *Int. J. Hybrid Intell. Syst.*, vol. 8, no. 4, pp. 169–184, Nov. 2011.

[139] H. Chu, W. Lu, and L. Zhang, "Application of artificial neural network in environmental water quality assessment," *J. Agricult. Sci. Technol.*, vol. 15, no. 2, pp. 343–356, 2013.

[140] L. Cantini and M. Caselle, "Hope4Genes: A hopfield-like class prediction algorithm for transcriptomic data," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.

[141] T. Washizawa, "Application of hopfield network to saccades," *IEEE Trans. Neural Netw.*, vol. 4, no. 6, pp. 995–997, Nov. 1993.

[142] N. Soni, E. K. Sharma, and A. Kapoor, "Application of hopfield neural network for facial image recognition," Blue Eyes Intell. Eng. Sci. Pub., Bhopal, India, Tech. Rep. A2816058119/19©BEIES.

[143] N. Soni, N. Singh, A. Kapoor, and E. K. Sharma, "Low-resolution image recognition using cloud hopfield neural network," in *Progress in Advanced Computing and Intelligent Engineering*. Singapore: Springer, 2018, pp. 39–46.

[144] K. S. Ray and D. Dutta Majumder, "Application of hopfield neural networks and canonical perspectives to recognize and locate partially occluded 3-D objects," *Pattern Recognit. Lett.*, vol. 15, no. 8, pp. 815–824, Aug. 1994.

[145] D. O. Albanez, S. F. da Silva, M. A. Batista, and C. A. Z. Barcelos, "Images segmentation using a modified hopfield artificial neural network," *Proc. Ser. Brazilian Soc. Comput. Appl. Math.*, vol. 6, no. 1, pp. 1–7, 2018.

[146] F. Sabahi, M. O. Ahmad, and M. N. S. Swamy, "Hopfield network-based image retrieval using re-ranking and voting," in *Proc. IEEE 30th Can. Conf. Electr. Comput. Eng. (CCECE)*, Apr. 2017, pp. 1–4.

[147] X. Yang, L. Zhao, G. M. Megson, and D. J. Evans, "A system-level fault diagnosis algorithm based on preprocessing and parallel Hopfield neural network," in *Proc. 4th IEEE Workshop RTL High Level Test.*, Nov. 2003, pp. 189–196.

[148] 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.

[149] H. Yang, B. Wang, Q. Yao, A. Yu, and J. Zhang, "Efficient hybrid multi-faults location based on hopfield neural network in 5G coexisting radio and optical wireless networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1218–1228, Dec. 2019.

[150] B. Wang, H. Yang, Q. Yao, A. Yu, T. Hong, J. Zhang, M. Kadoch, and M. Cheriet, "Hopfield 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.

[151] Y. Watanabe, N. Mizuguchi, and Y. Fujii, "Solving optimization problems by using a hopfield neural network and genetic algorithm combination," *Syst. Comput. Jpn.*, vol. 29, no. 10, pp. 68–74, Oct. 1998.

[152] R. A. Silver, "Neuronal arithmetic," *Nature Rev. Neurosci.*, vol. 11, no. 7, pp. 474–489, Jul. 2010.

[153] 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.

[154] D. Thain, *Introduction to Compilers and Language Design*. Morrisville, NC, USA: Lulu, 2019.

[155] L. Wang, S.-R. Lu, and J. Wen, "Recent advances on neuromorphic systems using phase-change materials," *Nanosc. Res. Lett.*, vol. 12, no. 1, p. 347, Dec. 2017.

[156] G. Tanaka, T. Yamane, J. B. Héroux, 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.

[157] V. Saxena, X. Wu, I. Srivastava, and K. Zhu, "Towards neuromorphic learning machines using emerging memory devices with brain-like energy efficiency," *J. Low Power Electron. Appl.*, vol. 8, no. 4, p. 34, 2018.

[158] P. Date, C. D. Carothers, J. A. Hendler, and M. Magdon-Ismail, "Efficient classification of supercomputer failures using neuromorphic computing," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 242–249.

[159] D. Ielmini and S. Ambrogio, "Emerging neuromorphic devices," *Nanotechnology*, vol. 31, no. 9, Feb. 2020, Art. no. 092001.

[160] S. Furber, "Large-scale neuromorphic computing systems," *J. Neural Eng.*, vol. 13, no. 5, Oct. 2016, Art. no. 051001.

[161] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, Nov. 2019.

[162] 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 WS$_2$ 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 engineering, and the master's degree in information technology from the University of Wolverhampton, in 2014 and 2015, respectively. He is currently pursuing the Ph.D. Research degree with the University of Glasgow, Glasgow, U.K. His research interests include low-power neuromorphic sensor fusion, neuromorphic computing, embedded systems, 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 Affiliate Academic with the University of Glasgow, U.K. He is also the Founder and the Chair of the IEEE Communication Society, Oman. He was nominated for the Royal Academy of Engineering 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 University 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.

HADI HEIDARI (Senior Member, IEEE) received the Ph.D. degree. He is currently an Assistant Professor (Lecturer) with the School of Engineering, University of Glasgow, U.K. He leads the Microelectronics Laboratory and the Research, includes developing microelectronics and sensors for neurotechnology devices. He has authored more than 90 articles in tier-1 journals and conferences. He is a member of the IEEE Circuits and Systems Society Board of Governors (BoG) and the IEEE Sensors 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 č1m 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 Ă 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 Professor of communication systems with the School of Engineering, University of Glasgow. He is an Affiliate Professor with The University of Oklahoma, 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 Glasgow. 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 č6 million in sponsored research grants and contracts. He has supervised 40+ 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 finalist 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.

QAMMER H. ABBASI (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics 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 (Assistant Professor) with the School of Engineering, University of Glasgow, U.K., and a Researcher Investigator with The Scotland 5G Center. He has contributed 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.