Proceedings of to the 9th Scientific Conference of University of Garmian:
Pure science and Technology Applications (SCUG-PSTA-2022).
Kalar (46021) / Sulaymaniyah (26-27 Oct.2022)
(Special issue for Passer journal of basic and applied sciences)
Paper No. 07, 92-106

Continuous Proteretic Hopfield Neural Network in Walsh-based Distributed Storage

Aysar Abdulkhaleq Abdulrahman[1*]
[1*] **Department of Computer, College of Science, University of Sulaimani, Sulaimani, Iraq**
Aysser.abdulrahman@univsul.edu.iq

ABSTRACT

A neural network is a series of algorithms that endeavour to recognize underlying relationships in a set of data through a process that mimics the operations of a human brain. It may have uses in many different real-life applications, such as: speech and voice recognition, eCommerce, cybersecurity, and others. The network convergence time is the one of the most important parts of neural networks, which affects the performance of neural network applications. Convergence of the neural network contributes to the process of determining the optimal number of training iterations required to produce the fewest number of errors. In this paper, a specific method based on the mathematical property found in physical systems called "proteretic" is presented. Three learning methods (standard Hopfield, hysteretic Hopfield, and modified proteretic Hopfield) are applied to the Walsh-based distributed memory application. It mathematically and practically demonstrated and approved that using the modified proteretic method causes the network to reach convergence faster than other methods. It's approved that using the proteretic property with the Walsh-based memory enhances the performance of the storage by accelerating the network's convergence relative to other neural network operations.

KEYWORDS: Hysteresis, porteresis, walsh-based memory, activation function, convergence.

1 INTRODUCTION

Neural networks are utilized to simulate the fundamental functions of the human brain and are inspired by the way the brain interprets information. It is used to solve a variety of real-time problems due to its ability to perform computations rapidly and its quick response time. Layers are the typical structure for neural networks. The "activation function" is found in each of the numerous interconnected "nodes" that make up a layer. The "input layer" presents patterns to the network and communicates with one or more "hidden layers," where the real processing is carried out using a system of weighted "connections."

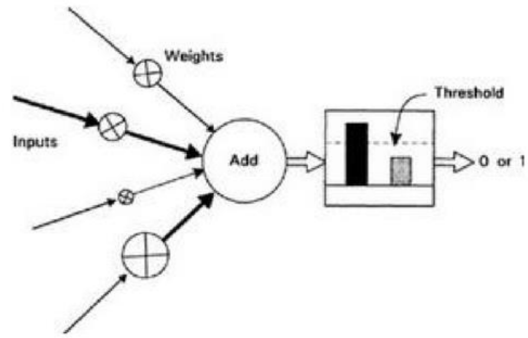
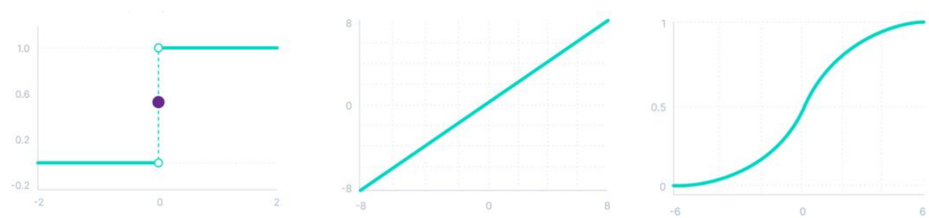


Fig.1: A simple artificial neuron [1]

It develops an adaptive system that computers utilize to continuously learn from their mistakes and get better. Thus, artificial neural networks aim to tackle complex tasks with higher precision, such as document summarization, facial recognition, and optimizing [2]. A topology refers to the way in which nodes are interconnected. In feed-forward neural networks, each node in one layer is connected to each node in the next layer. In recurrent neural networks, backward connections are permitted so that the output of one node can be fed into an earlier layer for the subsequent model iteration, which is useful when a node's previous output has an effect on its subsequent output. An essential component of a neural network's design is its activation functions. The choice of activation function in the hidden layer determines how efficiently the network model learns the training dataset. There are many kinds of functions that can be used as activation functions for the neural network systems depending on the application of the network, such as the binary step function, the linear function, or the logistic function, as shown in figure 2 below:



binary step function　　liner function　　logistic function

Fig.2: Samples of neural network activation functions

Neural network models come in a wide range, each of which can be used for several tasks and applications. Each model, however, has advantages and disadvantages [3], [4]. There are three distinct types of learning in neural networks, which are supervised learning, unsupervised learning, and reinforcement learning. In the supervised learning, there are training inputs and desired outputs. Every iteration model output is compared with the desired output and send back into the model until the output matches the desired output. Unsupervised network learning has no desired output, and the network learn on its own. The inputs are classified into classes during training, and based on similarities to other patterns, it may determine which class the input belongs to. The reinforcement learning achieves the best results from both supervised learning and unsupervised learning. It’s like learning based on the critique information [5], [6].

2 WALSH-BASED DISTRIBUTED MEMORY APPLICATION

Let x=(x1, x2, .., xn) and y=(y1, y2, ..., yn) be two n-dimensional vectors; and let us assume that it is feasible to superimpose many patterns, each with their own references, on the same memory media. So,

$$M = \sum_{k=1}^{m} x_k * w_k \tag{1}$$

Where w_k is a walsh function, and m is the number of patterns. Memory M is called the walsh based distributed associative memory[7]. In order to retrieve a specific item x_j from the memory, the specific item x_j must first be retrieved from its corresponding Walsh function w_j in the manner described in the following:

$$\widetilde{\mathbf{x}}_j = \mathbf{x}_j + \sum_{k \neq j} \mathbf{x}_k * \mathbf{W}_j * \mathbf{W}_k \tag{2}$$

Despite this, (x_j) is retrieved to $(\bar{x}_j)$ as a result of interference or crosstalk $(\sum_{k\neq j} x_k * w_j * w_k)$ from the other items that re superimposed on it. To retrieve the proper elements, this interference or crosstalk term must be minimized. In processes involving pattern matching, the correlation factors between the input data x and the desired data x_j must be evaluated as follows:

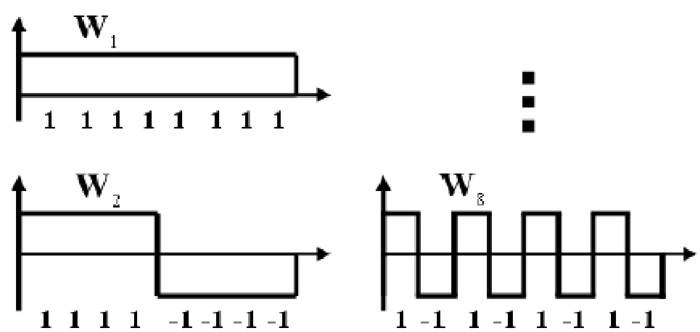


Fig.3: Walsh orthogonal basis functions in 8-Dimension

$$\mathbf{x} \cdot \mathbf{x}_j = \left[WAL(\mathbf{x} * \mathbf{M}) \right]_j \ \text{for} \ j = 1, \cdots, m \tag{3}$$

It is interesting to note that the fast Walsh transform (which requires just $n\text{Log}_2 n$ operations where n is the dimension of the data) enables the evaluation of all correlations in parallel with a single transform operation. Consequently, a Walsh-based distributed associative memory may store *m* patterns in a single datum storage space using Walsh encoding for each pattern and has *m* times the matching speed of Euclidean distance computation [8].

3 HOPFILED NEURAL NETWORK

The Hopfield neural network was invented in 1982 by Dr. John J. Hopfield. It has one or more fully interconnected recurrent neurons in a single layer, as shown in figure 4.

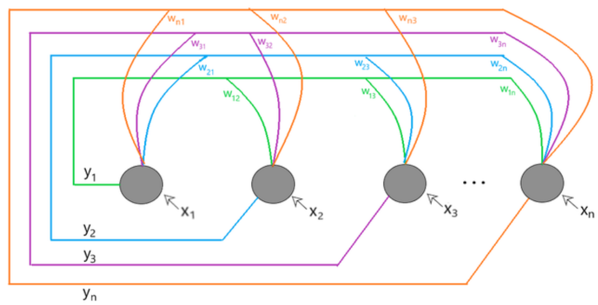


Fig.4: Basic Hopfield Network Architecture

Auto-association and optimization are the two applications that make frequent use of the Hopfield neural network, which methodically matches any input with one of the data vectors or patterns that were previously kept in its memory. Every single data pattern that is saved in the network makes a contribution to the weight matrix [9] [10].

For the purpose of characterizing the network, a Lyapunov function is applied. The energy function is computed for each iteration of the network based on the input that is provided. An output is generated by the network whenever the total amount of energy reaches its lowest possible value. The following is a description of the energy function for the Hopfield neural network:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}X_iX_j - \sum_{i=1}^{N} I_iX_i \tag{4}$$

Where E represents energy, N represents the number of neurons, W represents the connections between neurons, X represents the input, and I represents an input bias. The movement of a ball in a parabola can be used to illustrate the energy function; the ball should always start at the top of the parabola and finish up at the bottom, which is the local minimum. It is possible to update the output of the units in either a parallel fashion using synchronous methods or in an asynchronous fashion using asynchronous strategies (sequential). The manner in which the scheme is brought up to date may have an effect on the network's convergence [11]. The continuous Hopfield is based on the premise that each neuron is represented by an operational amplifier that contains either a resistor network or a capacitor network. Two different signals can enter each cell. The first factor is known as constant external input, and it is denoted by the notation I_1, I_2,...., I_N. The other input comes from the feedback of an additional operational amplifier. The outputs of the amplifiers (a_1, a_2,......, a_N) are connected to the resistors (R), which act as a load. A negative input to a neuron can be obtained by selecting the output of an amplifier in its inverted state and applying that signal to the neuron. Using Kirchhoff's law, the differential function for the electrical circuit is derived as:

$$C\frac{dx_i(t)}{dt} = \sum_{j=1}^{N} T_{i,j}\, a_j(t) - \frac{x_i(t)}{R_i} + I_i \tag{5}$$

where,

$$|T_{i,j}| = \frac{1}{R_{i,j}} \tag{6}$$

$$\frac{1}{R_i} = \frac{1}{\rho} + \sum_{j=1}^{N} \frac{1}{R_{i,j}} \tag{7}$$

$$a_i = f(x_i) \tag{8}$$

In addition, the circuit is symmetric, therefore $T_{i,j}=T_{j,i}$,and the transfer function of the amplifier, $a_i= f(x_i)$, is a sigmoid function. By multiplying both sides of equation 7 by R_i :

$$R_i C\frac{dx_i(t)}{dt} = \sum_{j=1}^{N} R_i T_{i,j}\, a_j(t) - x_i(t) + R_i I_i \tag{9}$$

The last equation can transfer to a standard neural network notation as:

$$\varepsilon\frac{dx_i(t)}{dt} = -x_i(t) + \sum_{j=1}^{N} W_{i,j}\, a_j(t) + b_i \tag{10}$$

where,

$$\varepsilon = R_i C \quad W_{i,j} = R_i T_{i,j} \text{ , and } \quad b_i = R_i I_i \tag{11}$$

Finally, by transforming equation 10 into vector form, we will obtain:

$$\varepsilon\frac{d}{dt}x(t) = -x(t) + W\, a(t) + b \tag{12}$$

Where W is the weight matrix that represents the synaptic connections, b is the biases, a(t) = Ø(x(t)) is the nonlinear input/output activation levels, and x(t) is a vector of neuron activation levels [12], [13].

4 HYSTERETIC HOPFILED NEURAL NETWORK

The word "deficiency" or "lagging behind" originates from the Greek word "hysteresis," which gives rise to the English word "hysteresis." The analogy of a thermostat is the most effective and straightforward way to explain what hysteresis is. For instance, when we want to keep the temperature in our house at 80 degrees all the time, we have to adjust a thermostat so that it reads 80 degrees. It does not mean that the furnace will turn "on" or "off" whenever the thermometer stops reading 80 degrees, because doing so would cause the furnace to work harder than it is designed to. Instead of doing that, the furnace will begin to heat the home when the temperature falls below 75 degrees Fahrenheit, and it will remain "on" until the thermostat registers 85 degrees, at which point it will automatically shut off. In other words, hysteresis happens when an object does not return to its previous form after the removal of a force [14].

Therefore, we can argue that hysteresis is a system with memory because the current output of a hysteretic system is controlled by both the current input and the input's history. A hysteresis of a neural network can be expressed in a variety of ways, such as a binary hysteresis, which is illustrated in figure 5 as follows:

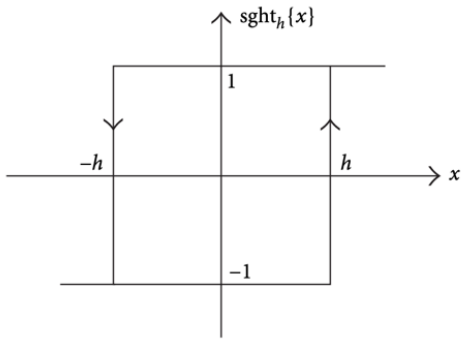


Fig.5. Binary Hysteretic function [15]

where point (a) indicates the upper trip point and point (b) indicates the lower trip point. Then, a neuron will only change its state from 0 to 1 when the total input to the neuron is greater than (a), and it will only change its state from 1 to 0 when the total input is less than (a) (b). Aside from that, it continues to maintain the same state.

$$V_i(t+1) = \begin{cases} 1 & , \ if \ u_i(t) > a \\ 0 & , \ if \ u_i(t) < b \\ no\ change & , if \ b \leq u_i(t) \leq a \end{cases} \quad (13)$$

This hysteretic property increases the rate of successful associative learning and may lead to an error that prevents the states of neurons from being altered [16]. A series of curves is another way to explain what is meant by the word "hysteresis." Bharitkar and Mendel came up with the idea of a flexible and hysteretic Hopfield network back in the year 2000 [17]. As can be seen in figure 6, below, there are four adjustable parameters in the function, which allow the system to be adjusted so that it performs at its highest level possible for the particular application:

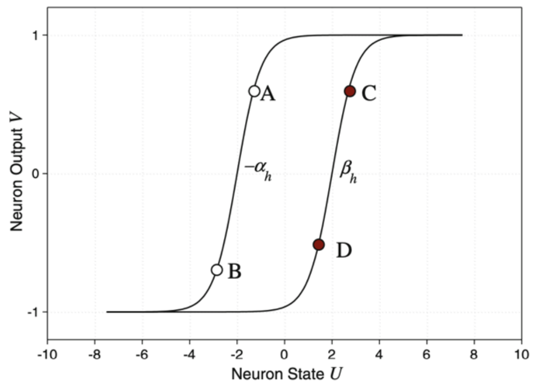


Fig.6: Continuous hysteresis function [18]

Figure 6 depicts the form of the sigmoid hysteretic function, in which the activation function represents two sigmoid functions, each with two center parameters.

$$f(s) = \begin{cases} \left(1 + e^{-c1(s-a)}\right)^{-1}, & if\ s(t+1) > s(t) \\ \left(1 + e^{-c2(s+a)}\right)^{-1}, & if\ s(t+1) < s(t) \end{cases} \qquad (14)$$

If the value of the subsequent input has increased after an initial input, then the output will continue to move upward along the right curve. If, on the other hand, the value of the subsequent input is lower than the value of the previous input, then the output will shift to the left curve [19]. The Hopfield neural network has had hysteretic neurons added to it, and these neurons are now being successfully used in a wide variety of combinational optimization problem-solving applications. Hysteresis can be defined as the slowing of an effect when acting forces on a body are altered, and it has a wide range of applications. The previously noted issue with the energy function is remedied by the hysteretic neuron, which also makes it possible for oscillations in the output to arrive to a globally optimal solution. Making it possible for there to be oscillations cuts down on the number of inappropriate solutions in the space [20], [21].

The Hopfield network has a scalar value called the energy, E, of the network that is associated with each state of the network. This value is what is used to prove that the network is stable. In the Hopfield neural network, the following constitutes the energy function:

$$\mathrm{E} = -\frac{1}{2}\sum_i \sum_j W_{ij} V_i V_j - \sum_i \theta_i V_i \qquad (15)$$

Where θ_i is the value that represents the network's threshold. If we assume that the network has a threshold of zero, then the energy function is going to be as follows:

$$\mathrm{E} = -\frac{1}{2}\sum_i \sum_j W_{ij} V_i V_j \qquad (16)$$

The energy function is utilized to demonstrate the stability of recurrent networks. In other words, the network is in a stable state if a state is a local minimum in the energy function [22].

5 LITERATURE REVIEW

A neuron activation function based on the hysteresis phenomenon of physical systems is proposed in this paper in order to form the hysteretic Hopfield neural network. In this work, the Lyapunov stability for this new model has been proven, then applied to the N-queen network problem. The advantage of hysteresis is demonstrated by showing increased frequency of convergence to a solution, when the parameters associated with the activation function are varied [23]. In 2010, the hysteretic neuron with predictive hysteresis was proposed, and the effectiveness of employing these neurons in analog to digital hardware network device via the Hopfield neural network. Hysteretic neurons have been widely used in neural networks, where they have been relatively successfully exploited in the solution of combinatorial optimization problems. This finding has implications in a range of many different applications, including neural networks applications. It was demonstrated that predictive hysteresis property enhanced the speed of both hysteretic and non-hysteretic systems without reducing the performance and the accuracy of those systems [24]. In [21] a design and simulation (using VPIphotonics) of an all-optical proteretic bi-stable device was reported for the very first time. The proteresis is a reversal of the hysteresis phenomenon, and it possesses an intriguing property in that it can, by decreasing the feed-back delay, cause an increase in the oscillation frequency of a feed-back system that possesses a relaxation dynamic. A calculation of the bi-stable device parameters, as well as simulations of both the theoretical device and the all-optical device, are provided here. Also, in 2020, a new semiconductor ring laser-based optical proteretic system is demonstrated. The design was carried out using a hysteretic feedback system with a constraint feedforward path. Also, other parameters were studied in order to accomplish proteolysis. The study showed that only specific circumstances allow for the proteretic effect; Otherwise, the system functions like a typical hysteretic device [25].

6 WALSH-BASED MODIFIED PROTERETIC HOPFILED NEURAL NETWORK

A distributed memory based on the Walsh function is able to store many patterns using Walsh encoding in a single storage region. In comparison to other types of memories, it has distinct advantages in terms of reduced storage and quick recall. When compared to proteresis, hysteresis can be seen as "that which comes after," while proteresis can be understood as "that which comes before." The distinction between proteresis and hysteresis is that, in the case of proteresis, the output will move upward along the left curve whenever there is a positive difference between the two successive inputs. On the other hand, when there is a negative difference between successive inputs, the output will move in the right direction, following the right curve. As shown in equations 17, 18, and figure 7 below, the proteretic activation function is created by inverting the directions of the hysteretic function in order to decrease the delay time for changing the neuron states from 1 to 0 or from 0 to 1. This ultimately results in a reduction in the amount of time required for the network to converge [26].

$$f(s) = \begin{cases} \left(1 + e^{-c1(s+\alpha)}\right)^{-1}, & \text{if } s(t+1) \geq s(t) \\ \left(1 + e^{-c2(s-\alpha)}\right)^{-1}, & \text{if } s(t+1) < s(t) \end{cases} \tag{17}$$

$$f(s) = \begin{cases} \left(1 + e^{-c1(s+\alpha)}\right)^{-1}, & if\ s(t+1) > s(t) \\ \left(1 + e^{-c2(s-\alpha)}\right)^{-1}, & if\ s(t+1) < s(t) \\ No\ change, & if\ s(t+1) = s(t) \end{cases} \tag{18}$$

where c1, c2, s, and α are adjustable variables that allow for optimal learning convergence tuning of the neural network. We experimented with numerous tuning variables throughout this project until we achieved the desired network outputs. As a result, adding the proteretic property to the Walsh-based distributed memory device architecture improves its performance by decreasing the convergence time.

7 FEASIBILITY OF OUR ASSUMPTION

We predicted that proteresis would cause the neural network system to have a substantially lower convergence rate than the standard Hopfield and hysteretic Hopfield network while maintaining its performance. Our assumption is based on two factors:

The first is based on the Lyapunov stability rule. The energy function of the hysteretic Hopfield neural network is identical to the proteretic Hopfield neural network, which was introduced and described by Bharitkar and Mendel in 2000 as follows [17]:

$$E(y_1, y_2, \dots, y_N) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij} y_i y_j + \sum_{j=1}^{N}\frac{1}{R_j}\int_0^{y_j} \phi_j^{-1} x\left[y_j\left(x_j|\dot{x}_j\right)\right] dy_j - \sum_{j=1}^{N} I_j y_j \tag{19}$$

According to Lyapunov, there are certain conditions that must be met in order to determine whether or not a system is stable. This condition, which is the derivative of the energy function that represents the system, must retain negative values for all time with respect to time, which indicates that:

$$\frac{dE(y_1, y_2, \dots, y_n)}{dt} \leq 0 \tag{20}$$

To demonstrate the stability of the hysteretic function defined by Bharithar and Mendel, it must be demonstrated that the value of the energy function's derivative with respect to time never exceeds zero. The expression for the derivative of the energy function is:

$$\frac{dE}{dt} = -\sum_{i=1}^{N} C_j \frac{dy_j}{dx_j}\left(\frac{dx_j}{dt}\right)^2 \tag{21}$$

Since the square of any number is always positive, $(dx_j/dt)^2$ is always positive. Also, because capacitance is always positive, C_j is always positive. Furthermore, (dE/dt) polarity is determined by the polarity of the difference between two subsequent output values of neuron j divided by the polarity of the corresponding two input values, that can be stated as the following:

$$\frac{dE}{dt} \Rightarrow \frac{\triangle y_j}{\triangle x_j} \tag{22}$$

When taking into consideration the hysteretic activation function, the output moves upward along the right curve when the difference between successive inputs yields a negative value, $\triangle x < 0$, and moves downward along the left curve when the difference between successive inputs yields a positive value, $x > 0$, as depicted in figure 7. In other words, the output moves in the opposite direction of the inputs when the difference between successive inputs yields a positive value. Transition from the left to right curves when the input is rising, $\triangle x > 0$ and $\triangle y < 0$. In this instance, $(\triangle y/\triangle x)$ is less than zero and (dE/dt) is greater than zero, indicating that the energy is growing, which contradicts the Lyapunov stability rule. When the input is decreasing $\triangle x < 0$ and $\triangle y > 0$, the right curve changes to the left curve. In this instance, $(\triangle y/\triangle x)$ is less than zero and (dE/dt) is greater than zero, indicating that the energy is rising, which also violates the Lyapunov stability criterion. When there is no transition between the curves, $(\triangle y/\triangle x) > 0$ and $(dE/dt) < 0$, the Lyapunov stability criterion is not violated. Thus, it is important to note that the proof for the Hysteretic Hopfield Network has failed. As depicted in Figure 7, the output of the proteretic activation function rises along the left curve when the difference between successive inputs provides a negative value, $\triangle x > 0$, and falls along the left curve when the difference between successive inputs yields a positive value, $\triangle x < 0$. When the input is decreasing $\triangle x < 0$ and $\triangle y < 0$, the transition from the left curve to the right curve happens. In this instance, $(\triangle y/\triangle x) > 0$ and $(dE/dt) < 0$; hence, the Lyapunov stability condition is not violated. When the input is growing, $\triangle x > 0$ and $\triangle y > 0$ a transition from the right curve to the left curve happens. In this instance, $(\triangle y/\triangle x) > 0$ and $(dE/dt) < 0$; hence, the Lyapunov stability condition is not violated. When there is no transition between the curves, $(\triangle y/\triangle x) > 0$ and $(dE/dt) < 0$, and the Lyapunov stability criterion is not violated. Consequently, it is important to note that the proof is supported by the proteretic Hopfield network.

Visual observation of the proteretic and hysteretic curves is the second basis for our hypothesis that reversing the direction of the hysteresis curves will result in faster convergence rates. Consider the example of input expansion. In the case of hysteretics, the output rises along curve B, whereas it rises along curve A in the case of proteretics.

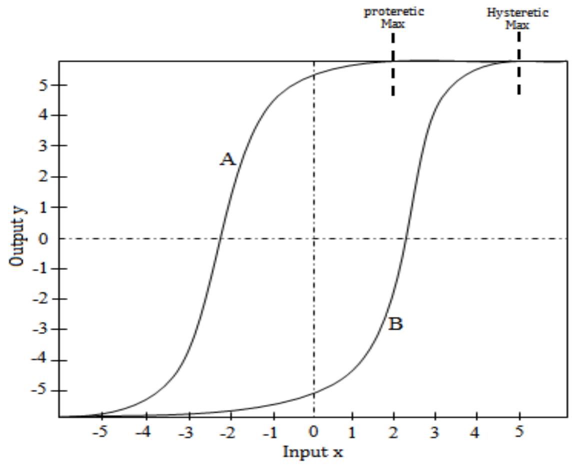


Fig.7: Hysteretic and proteretic curve

As demonstrated in Figure 7, the maximum output value is obtained more quickly in the proteretic situation than in the hysteretic one. Thus, the change in resultant velocity is proportional to the difference between the two values, proteretic maximum and hysteretic maximum [26].

8 SIMULATION RESULTS

In the first part of the simulation, we used a memory pattern group consisting of 12 neurons, as shown below: X_1 = [1 1 1 1 -1 1 -1 1 -1 1 1 1], X_2 = [1 1 -1 1 1 1 1 -1 1 -1 -1 1], and X_3 = [1 1 -1 1 -1 -1 1 1 -1 -1 1 1]. Using the Hebb rule, the connection weights, W, of the network for the group patterns consisting of 12 neurons are:

$$W = \begin{bmatrix} 1 & -1 & 0 & 0 & 2 & 1 & -3 & 1 & 0 & 2 & 1 & 0 \\ 1 & 1 & -2 & -1 & 1 & 1 & 1 & 1 & 0 & 0 & 2 & 3 \\ 0 & 1 & 1 & 1 & 2 & 2 & -2 & 3 & 0 & 1 & 1 & 0 \\ 0 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 3 & 1 & -3 & 0 \\ 3 & 0 & 1 & 2 & 0 & 2 & 2 & 0 & -3 & 2 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & 1 & 2 & 1 & 2 & 1 & 1 & 1 & 3 \\ 1 & -1 & -2 & 1 & 1 & 2 & 1 & 1 & 0 & 2 & 1 & 2 \\ 1 & 0 & 1 & 3 & 1 & 2 & 1 & 1 & 0 & 2 & 1 & 0 \\ -3 & 1 & 1 & 0 & 1 & 2 & 1 & 0 & 1 & 2 & 1 & 1 \\ -2 & 1 & 1 & 0 & 3 & 0 & 1 & -1 & 2 & 2 & 2 & 3 \\ 1 & 0 & 3 & 1 & 1 & 1 & 1 & -1 & 2 & -3 & 1 & 2 \end{bmatrix} \quad (23)$$

As shown in Tables 1, we first compared the results of the discrete and continuous portions of the functions by employing five distinct random initial values in our experiment. We utilized the asynchronous approach to update the output of the units, which implies that only one unit was updated at a time.

All activation functions, including the normal, hysteretic, and modified activation functions, were given the same initial values. Then, we compared the outcomes and performance of the functions by determining the number of iterations required to obtain convergence for each model.

No.	Initial Value	Network Iteration Number		
		Normal Neural Network	Hysteretic Neural Network	Modified Hysteretic Neural Network
1	[0101 1100 1010]	14	22	8
2	[1100 1010 0011]	11	19	7
3	[1010 1111 1100]	9	13	5
4	[1000 1100 1110]	16	23	11
5	[1110 0001 0101]	13	18	10

Table1: Number of iterations (convergence speed)

As indicated in Table 1, our test accepts a variety of initial values as input. With 12 neurons, there are 4096 possible outcomes. Table 1 displays the output of the network with simply five random initial values as input. The initial values were selected at random and listed in the first column. The second column presents the number of iterations of the normal Hopfield neural network. The third column presents the number of iterations of the hysteretic Hopfield neural network. The fourth column presents the number of iterations of the modified proteretic Hopfield neural network. Table 1 demonstrates that the modified proteretic Hopfield network enabled the initial values to converge faster than the other networks, which is the best outcome compared to the normal Hopfield network and the hysteretic Hopfield network. For instance, the simulation outcome of the first initial value [010111001010] is illustrated in figure 8:

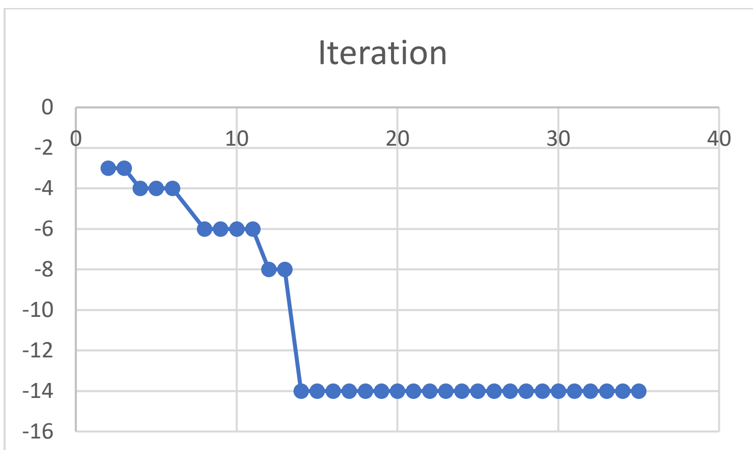


Fig.8-a: Output of the Hopfield function with initial value [010111001010]

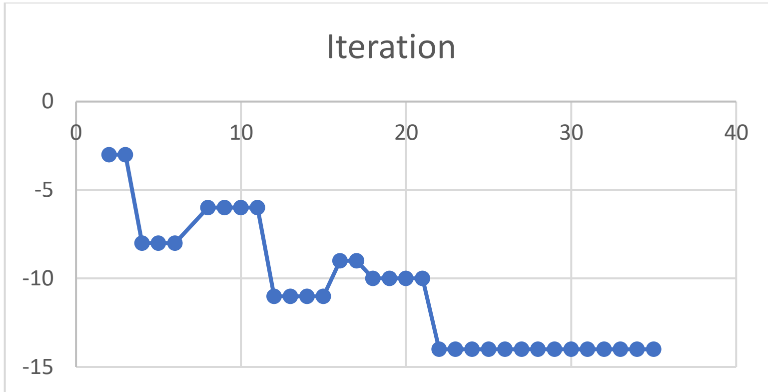


Fig.8-b: Output of the hysteretic Hopfield with initial value [010111001010]

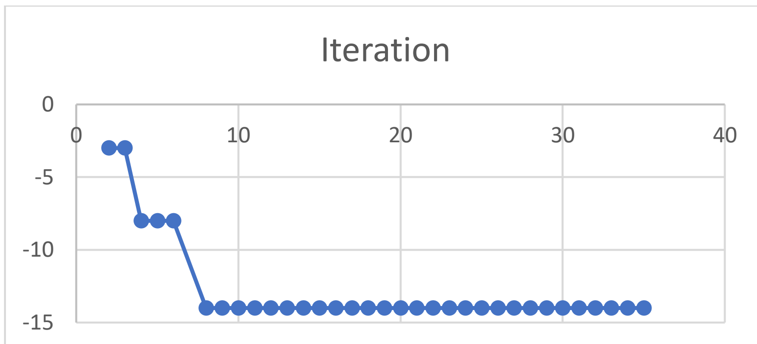


Fig.8-c: Output of the modified Proteretic function with initial value [010111001010]

In the second part of the simulation, and for the purpose of enhancing our experiment, we applied our test to all the $(2)^{12}$=4096, $(2)^{14}$=16384, and $(2)^{16}$=32768 possibilities, and by calculating the average number of iterations for all possible outcomes, as shown in table 2:

Number of the neurons	Average of all the possibilities		
	Normal Neural Network	Hysteretic Neural Network	Modified Proteretic Neural Network
12	13.6622	12.8000	11.1398
14	15.3766	15.0899	13.3339
16	16.4784	16.8891	15.3001
Average	**15.1724**	**14.92633**	**13.25793**

Table 2: Average of all the input possibilities (convergence speed)

Table 2 shows the results of our experiment, where the first column shows the number of the neurons, and the other three columns show the average of the number of iterations of all the possibilities. Although the difference between the three average outputs is not significantly height, increasing the number of neurons increases the disparity. In conclusion, according to Table 2 above, when compared to both the standard neural network and the hysteretic neural network, the output of the modified proteretic neural network produces the best results.

9 CONCLUSTION

Neural networks are computing systems with interconnected nodes that work much like the human brain's neurons. It has many different applications in our real lives. The performance of the network depends on some main factors, such as the number of network layers and the activation function. In this paper, we have introduced the Walsh-based distributed memory, and the convergence of the neural network. We have used three different learning activation properties (Hopfield neural network, hysteretic Hopfield neural network, and modified proteretic Hopfield neural network) to find the shortest convergence time. It has been demonstrated that combining the modified proteretic property with the Walsh-based memory improves the Walsh-based storage's performance in comparison to the standard Hopfield and hysteretic Hopfield neural networks.

REFERENCES

[1] K. Gurney, *An Introduction to Neural Networks*. London, England: CRC Press, 2018.

[2] S. Lynch, "Limit Cycles," in *Dynamical Systems with Applications using MATLAB\textregistered{}*, Cham: Springer International Publishing, 2014, pp. 223–247.

[3] S. O. Haykin, *Neural networks and learning machines: International edition*, 3rd ed. Upper Saddle River, NJ: Pearson, 2010.

[4] E. Goles and S. Mart\'\inez, "Automata Networks," in *Neural and Automata Networks*, Dordrecht: Springer Netherlands, 1990, pp. 15–37.

[5] C. C. Aggarwal, *Neural Networks and Deep Learning*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-94463-0.

[6] H. Xiao, "Stability of Hopfield Neural Networks with Time-Varying Delay," in *Advances in Neural Networks – ISNN 2009*, 2009, pp. 375–382.

[7] K. Kyung-A and O. Se-Young, *A Walsh-Based Distributed Associative Memory with Genetic Algorithm Maximization of Storage Capacity for Face Recognition*. 2003.

[8] S.-J. Han and S.-Y. Oh, "Enhanced Walsh function based distributed associative memory for pattern recognition," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006.

[9] K. Jin'no and T. Saito, "Analysis of a simple hysteresis network and its application for an effective associative memory," in *1993 IEEE International Symposium on Circuits and Systems*, 2005.

[10] E. Ansari Nirwan and Hou, "Hopfield Neural Networks," in *Computational Intelligence for Optimization*, Boston, MA: Springer US, 1997, pp. 27–45. doi: 10.1007/978-1-4615-6331-0_3.

[11] R.-N. Zhang Sheng-Rui and Ma, "Convergence Analysis of Discrete Delayed Hopfield Neural Networks," in *Advances in Neural Networks - ISNN 2006*, 2006, pp. 267–272.

[12] James P. Coughlin, “Neural computation in Hopfield networks and Boltzmann machines,” *Choice Reviews Online*, vol. 33, no. 02, pp. 33-0969-33–0969, Oct. 1995, doi: 10.5860/CHOICE.33-0969.

[13] Y. and M. J. and Z. Y. and Y. X. Li Jun and Diao, “Stability Analysis of Discrete Hopfield Neural Networks with Weight Function Matrix,” in *Advances in Computation and Intelligence*, 2008, pp. 760–768.

[14] I. D. Mayergoyz and Can Korman, *Hysteresis And Neural Memory*. Singapore, Singapore: World Scientific Publishing, 2019.

[15] G. Jeney, J. Levendovszky, L. Pap, and E. C. van der Meulen, “Adaptive near-optimal multiuser detection using a stochastic and hysteretic Hopfield net receiver,” *EURASIP J. Adv. Signal Process.*, vol. 2002, no. 12, 2003.

[16] C. Xiu and Y. Liu, “Hysteresis response neural network and its applications,” in *2009 ISECS International Colloquium on Computing, Communication, Control, and Management*, 2009.

[17] S. Bharitkar and J. M. Mendel, “The hysteretic Hopfield neural network,” *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 879–888, 2000.

[18] W. Zhang, T. Le, Y. Wu, Y. Jiang, and R. Fan, “Hysteretic chaotic neural network for crossbar switch problems,” *J. Eng. (Stevenage)*, vol. 2018, no. 16, pp. 1658–1661, 2018.

[19] Q. and W. Z. Wang Xiuhong and Qiao, “Solving Optimization Problems Based on Chaotic Neural Network with Hysteretic Activation Function,” in *Advances in Neural Networks – ISNN 2005*, 2005, pp. 745–749.

[20] N. Davoudzadeh, M. Tafazoli, and M. R. Sayeh, “All-optical proteretic (reversed-hysteretic) bi-stable device,” *Opt. Commun.*, vol. 331, pp. 306–309, 2014.

[21] P. Thangavel and D. Gladis, “Hysteretic Hopfield network with dynamic tunneling for crossbar switch and -queens problem,” *Neurocomputing*, vol. 70, no. 13–15, pp. 2544–2551, 2007.

[22] S. Bharitkar and J. M. Mendel, “The hysteretic Hopfield neural network,” *IEEE Trans. Neural Netw.*, vol. 11, no. 4, pp. 879–888, 2000.

[23] M. Sayeh and I. Calmese, “The Hopfield Neural Network with Predictive Hysteresis,” in *Intelligent Engineering Systems through Artificial Neural Networks Volume 18*, 2010. doi: 10.1115/1.802823.paper61.

[24] A. Syed, M. Tafazoli, N. Davoudzadeh, and M. R. Sayeh, “An all-optical proteretic switch using semiconductor ring lasers,” 2020.

[25] A. Aysar, “Investigation of Proteretic Hopfield Networks,” Doctoral dissertation, Southern Illinois University, Carbondale, 2013.