INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume3, Issue3)
Available online at www.ijariit.com

Study of Hopfield Neural Memory for Noisy Random Patterns Using Some Evolutionary Approach

Srishti Gangwar
M.tech
Department of Computer Science and Engineering
Sharda University
srishtigangwar93@gmail.com

T. P Singh
Ph.d
Department of Computer Science and Engineering
Sharda University
tp.singh@sharda.ac.in

***Abstract:** In this paper, a vigorous attempt has been made to study Hopfield neural network for storing and later recalling of random patterns using conventional hebbian learning rule and genetic algorithm. Storing of these patterns in the network is done using Hebbian learning rule followed by recalling of these patterns on presentation of distorted input patterns is done using both the methods i.e. Hebbian rule and genetic algorithm. The optimal weight matrix obtained is used to generate new weight matrices for the efficient recalling of prototype input patterns. Performance evaluation of the network is done on the basis of pattern recall with maximum noise present in patterns. The results thus obtained shows that the recall of these random patterns is more successful by using a genetic algorithm.*

***Keywords:** Hopfield Neural Network, Genetic Algorithm, Hebbian Learning Rule, Random Patterns, Pattern Storing and Recalling.*

1. INTRODUCTION

We all human being can recognize our friends easily whom we met a long time ago by just recalling a pertinent information about them. Numbers of examples demonstrate that a human brain can learn, understand and remember certain things partially, completely and not at all sometimes. As we all know no program yet can compete with the high-level intelligent functions of the human brain such as pattern recognition, categorization, and associative memory.

Artificial neural network or neural network is the network which processes information in a very similar manner to that of the human brain. A Neural Network is configured for pattern recognition or data classification, through a learning process.One such part of a neural network that deals with pattern storing and recalling is an associative memory. This memory allows its user to specify part of pattern or key and retrieve the values associated with that pattern even in the presence of noise. There are two types of associative memory in the neural network: auto-associative and hetero-associative. The difference between both memories lies in the there retrieved pattern. An auto-associative memory retrieves the same pattern B given an input pattern A, i.e., B = A. On the other hand, a hetero-associative memory retrieves the stored pattern B given an input pattern A such that B¹A.

A Hopfield neural network (HNN) is a simple recurrent type artificial neural network in which certain memories or patterns are stored in a very similar manner to that of the brain. John Hopfield proposed this model in early 1980. The Hopfield model is a type of auto-associative memory model. Even if some of the interconnections of nodes get changed, the recalled pattern is not badly corrupted, the network even though respond with the best result and this is all because of the stability of the system.

2. LITERATURE REVIEW

Over the past, various studies have been done to obtain an intelligent computer -simulation model that have same information processing capability to that of the human brain. Symbolic processing, fuzzy approach and artificial neural network are some of the major areas that include these approaches. From the previous studies, we found that among all the approaches, the associative memory is one of the best paradigms that best reflect the intention of emulating the associative nature of the human brain. Because of this reason, during the last years these memories have been widely studied, although many expressive paradigms have emerged, in many application fields, these memories show an advantage to others.

The Hopfield neural network (HNN) consists of n totally coupled units. In this network each unit is connected to all other units except itself. The connection between unit i and unit j is same as unit j and unit i as the weight w_{ij} is equal to weight w_{ji}, therefore the network is symmetrical and hence we interpret that there is a single bidirectional connection between both the units. Since there is no connection to itself of each state so it avoids the permanent feedback of its own state value. The network can provide the best result and the recalled memory is not badly corrupted even though some of the nodes (neurons) connections are disturbed. This network solves the problem of pattern storage and pattern recalls i.e. if we store a new pattern the model will return a pattern that closely resembles the new one from the set of stored patterns. For determining the weights of HNN which is an auto-associative neural network the most common and the simplest method is the Hebb rule [23]. The Hebb rule will provide the correct weights, and the response of the net, when tested with one of the training vectors, will be a perfect recall, if the input vectors are orthogonal i.e. uncorrelated but if the input vectors are not orthogonal, a portion of each of the target values will be included in the response. This method is called as cross talk.

The proposed stable pattern/memories are energy minima of Hopfield neural net. This type of network mostly gets trapped in non-optimal local minima close to the starting point, which is not desired. One of the most common problems with Hopfield network is that they settle in local minima, but for building a content addressable memory having local minima is good. The convergence to one of the stored pattern is not sure, but they are guaranteed to converge to local minima. As Hopfield network settles into the stable state via a completely disturbed algorithm that is why they cannot find a global solution. For searching global minima an evolutionary algorithm can be adopted and thereby reducing the problem of false minima. One of such evolutionary algorithm is a genetic algorithm which has been traditionally used in optimization [5&14]. In this, a random initial problem is generated for individuals, each of which represents a potential solution to a problem. In a problem, every member of the population's fitness as a solution is evaluated against some known criteria. Based upon the fitness of members, they are selected for reproduction from the off-springs of fit individuals a new generation of a potential solution is generated. Until the population converges to an acceptable solution this process of evaluation, selection and recombination are iterated.

Compression, approximation, steering are some of the techniques which use Hopfield neural network, but among all these Hopfield neural networks is commonly used for pattern recognition. In past literature work, we found that Hopfield neural network has been used for recognition of images, letters, alphabets, numerals, etc. for any distorted or disrupted pattern for already stored input patterns [4&31]. Hopfield neural network technique also found its application in the field of password authentication. Hopfield neural network overcomes the shortcomings of existing layered neural network system such as recall approximation and long training time and gives better accuracy and quick response time for password changes and registration [32].

In literature, a lot of work could be found [24-27] regarding using a genetic algorithm for evolution in the neural network. In a neural network, evolution has been introduced in three levels- architecture, connections, weights and learning rules. The present research focuses on exploring the evolution at connection weight levels. Previous work that focuses upon Hopfield neural network for the evolution of connection weights using GA has been found [28] and [17-18&29]. Interestingly, much of the references in literature is not found that focuses on using an evolutionary algorithm to train the network and then to retrieve the corrupted version of it. Handwritten character recognition is one of the common tasks realized by neural networks. In present work, an attempt is made by using a genetic algorithm to implement the evolution of weight matrices for both training and recall purpose simultaneously. The randomness from GA is minimized because it starts from the approximate optimum solution, instead of starting from the random solution; this is the main advantage of this approach. This is why the process of recall is more efficient and has relatively better error correction capability.

In the year 1997, Imadaet. al. proposed the application of evolutionary computation (genetic algorithm) to Hopfield neural network and found that using ternary chromosomes, both random weight matrix and over-loaded Hebbian weight matrix were successfully evolved [7]. Later, in the same year [8] they used a real encoded genetic algorithm to evolve random synaptic weights to store some number of patterns as an associative memory. Similar work related to use of Hopfield neural network can be found in [15, 17, 24 & 26]. In last recent year, we found that genetic algorithm is implemented to gradually develop the population of weight matrices for storing and recalling of patterns in HNN model and concluded that among all searching techniques for noisy input patterns genetic algorithm provides a better result than conventional Hebbian rule [28 & 29]. Then in the year 2011, Tomoya Shima and other author proposed partitioned Hopfield Neural Network (PHNN) to realize the memory mechanism of the human brain [34]. Another paper describing the storing and recalling of some prototype input patterns using genetic algorithm has been found [35] and the result was concluded that the recalls of the pattern are more successful in Hopfield neural network if the evolution of weight matrices is applied also for training purpose. We found in the later stages, Hopfield neural network of associative memory is used for storing and recalling of various types of input patterns and by using a different algorithm, the comparison between these algorithm has also been made to find the best algorithm [32 & 36-37].

3. METHODOLOGY

In this section, the experiment used to evaluate the performance of Hopfield neural network in the presence of Genetic Algorithm has been described for the set of input patterns taken into consideration for storing and recalling purpose.Herein this experiment, the Hebbian learning rule is used to store the patterns in the Hopfield neural network and two different algorithms, i.e. the Hebbian rule and the genetic algorithm, are used for recalling the patterns.

The model is implemented using MATLAB toolbox and command line coding procedure. Symmetric Hopfield Neural Network model consisting of various numbers of nodes is created. For training purpose, a set of 15 random patterns is generated for the experiment on a grid of 7x5 i.e. size 35 using bipolar values ‘+1’ for black and ‘-1’ for white. Using bipolar values the set of random patterns considered is represented in the form of series. The induction of noise is done by randomly flipping the bits in these patterns for recall purpose. The result of thus performed simulations has been tabulated and presented.

3.1. Hopfield Neural Network (HNN)

A Hopfield network is single-layered and recurrent network introduced by John Hopfield in the year 1982. In this network, the neurons are fully connected means each neuron in the network is connected to every other neuron. There is the connectivity weight w_{ij}, between two given neuron i and j which is symmetric i.e. $w_{ij} = w_{ji}$ with no self-connectivity i.e. $w_{ii} = 0$.

Updating Rule

- Assume N neurons = 1,…, N with values xi = ±1.
- For the node i, the update rule is given by:

If $h_i \leftarrow 0$ then $1 \leftarrow$ xi otherwise $1 \leftarrow x_i$

Where $h_i = \sum^{N}_{j=1} w_{ij}x_i + b_i$, called the field at i, with bi ϵ R a bias.

- Thus, $x_i \leftarrow \text{sgn}(h_i)$, where sgn(r) = 1 if r ≥ 0, and sgn(r) = -1 if r < 0.
- We put $b_i = 0$ for simplicity as it makes no difference to training the network with random patterns.
- We therefore assume $h_i = \sum^{N}_{j=1} w_{ij}x_j$.
- Now two ways are there to update the nodes:
    - **Asynchronously:** At each point in time, one node is updated which is chosen according to some rule or randomly.
    - **Synchronously:** Every time, update all nodes together.

- Biologically more realistic is asynchronous updating.

3.2. Hebbian Learning Rule

Hebbian learning rule was introduced by Donald Hebb in the year 1949 in his book 'The Organization of Behaviour'.The theory is also referred Hebb's postulate, Hebb's rule and cell assembly theory specifies how much the weight of the connection between two units should be increased or decreased in proportion to the product of their activation. This rule works well for the input patterns that are orthogonal or uncorrelated.

Hebb's principle can be described a method of determining how to alter the weights between model neurons. The two neurons when activated simultaneously then weight between them increases and if the neurons activate separately then weight between them decreases.

The formulaic description of Hebbian learning rule:

$w_{ij} = x_i x_j$

Where w_{ij} represents weight of the connection from neuron j to neuron i and x_i, the input to neuron i.

Another formulaic description is:

$$w_{ij} = \frac{1}{n}\sum_{k=1}^{m} x_i^k * x_j^k$$

Where *n* is the number of neurons and *m* is the number of training patterns.

3.3. Genetic Algorithm

A Genetic algorithm is a search technique to optimization and search problem to find the true or approximate solution.Commonly, the genetic algorithm works by creating a population with the group of individuals that are created randomly. Then the individuals in the population are evaluated. Then programmer provides the evaluation function and based on their performance the score is given to individuals. Based on the fitness value two individuals with higher fitness are selected, higher the fitness, the higher the chances of being selected. Reproduction of these individuals takes place to create one or more offspring, after which the offspring are mutated randomly. Depending on the needs of the programmer, the process continues until a suitable solution has been found or a certain number of generations have passed.

Following are the five main phases in GA algorithm:

a) Initialization- Create initial population.
b) Selection- Each member is then evaluated and we calculate a 'fitness' for that individual.
c) Crossover- Selection helps to keep the best individual in population by discarding the bad once.
d) Mutation- New individuals are created by combining aspects of selected individual.
e) Termination- Create new individual by making small changes at random to initial individual.

PSEUDO CODE OF THE PROCEDURE:

a) Set of 15 random patterns is presented and is stored in the network.

b) The parent weight matrix is determined for storing the input patterns.
c) By using the population generation technique new weight matrices are generated from the parent weight matrix.
d) By using crossover and mutation method the next generation of weight matrices is obtained.
e) Until the optimal solution is found the cycle of generating the new weight matrices with improved individual and restarting the search is repeated.

4. RESULT AND DISCUSSION

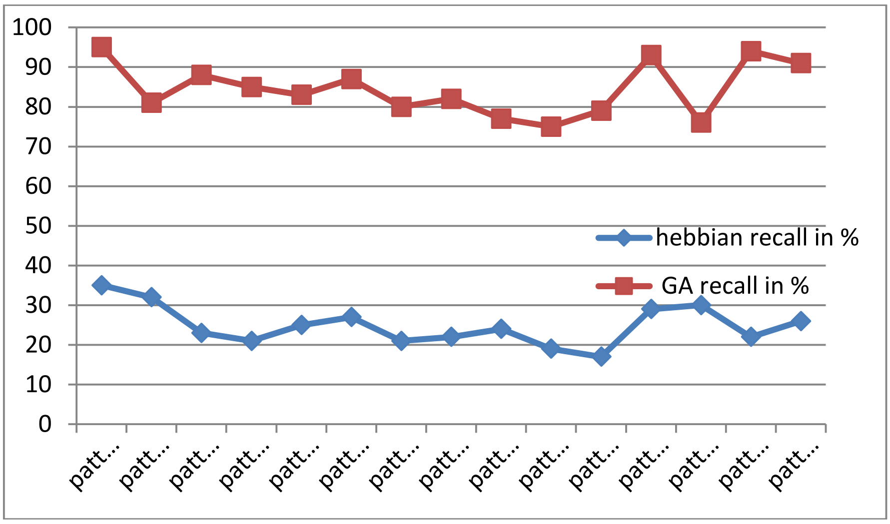
The results were demonstrated in this section, shows the significant difference between the performance of recalling success rate of genetic algorithm and conventional Hebbian rule for the random patterns with that have been stored in Hopfield neural network using Hebbian learning rule. From these results we concluded, for all cases recalling of any approximate random pattern through genetic algorithm outperformed the recalling of the same patterns through conventional Hebbian rule. With the evolution of weight matrices, the patterns are stored in the network and the recalling has been made by inducing error in the original input pattern, induction of noise is done by randomly flipping bits at different positions.

Table 2 below represent the results for recalling of stored input patterns for zero bit error i.e. when there is no noise induced in the original input patterns. The recalling for each pattern is done separately constituting 1000 runs using both the algorithms that include Hebbian rule and genetic algorithm.

Table 3, 4 and 5 below represents the results for recalling of the corresponding stored input patterns while these patterns are presented with the induction of noise. In the already stored input pattern, the noise was induced by flipping 1 bit, 2 bits, and 3 bits respectively. The bit/bits to be flipped to create noise in the patterns are randomly selected.

Table 1: Results of recalling of patterns when there no errors in the input patterns

No. of PATTERNS	Recalling success in (%)	
	Hebbian rule	**GA**
PATTERN 1	35	95
PATTERN 2	32	81
PATTERN 3	23	88
PATTERN 4	21	85
PATTERN 5	25	83
PATTERN 6	27	87
PATTERN 7	21	80
PATTERN 8	22	82
PATTERN 9	24	77
PATTERN 10	19	75
PATTERN 11	17	79
PATTERN 12	29	93
PATTERN 13	30	76
PATTERN 14	22	94
PATTERN 15	26	91



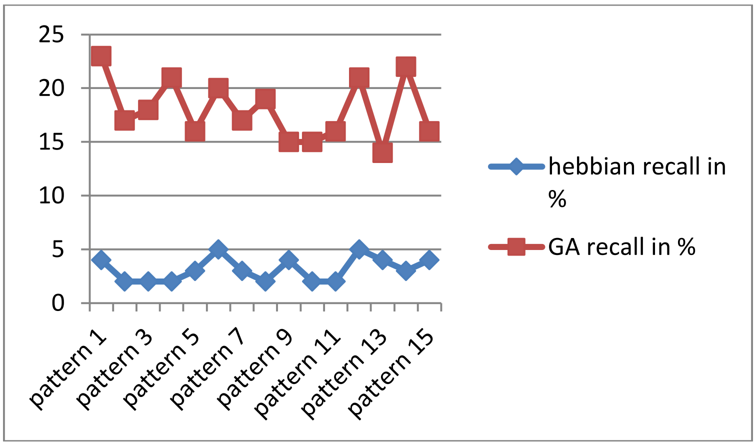
No. of patterns

Fig 1: Graph representing the recalling accuracy of patterns in (%) where there is no noise induction

Table 2: Results of recalling of patterns when there 1-bit errors in the input patterns

No. of PATTERNS	Recalling success in (%)	
	Hebbian rule	**GA**
PATTERN 1	4	23
PATTERN 2	2	17
PATTERN 3	2	18
PATTERN 4	2	21
PATTERN 5	3	16
PATTERN 6	5	20
PATTERN 7	3	17
PATTERN 8	2	19
PATTERN 9	4	15
PATTERN 10	2	15
PATTERN 11	2	16
PATTERN 12	5	21
PATTERN 13	4	14
PATTERN 14	3	22
PATTERN 15	4	16

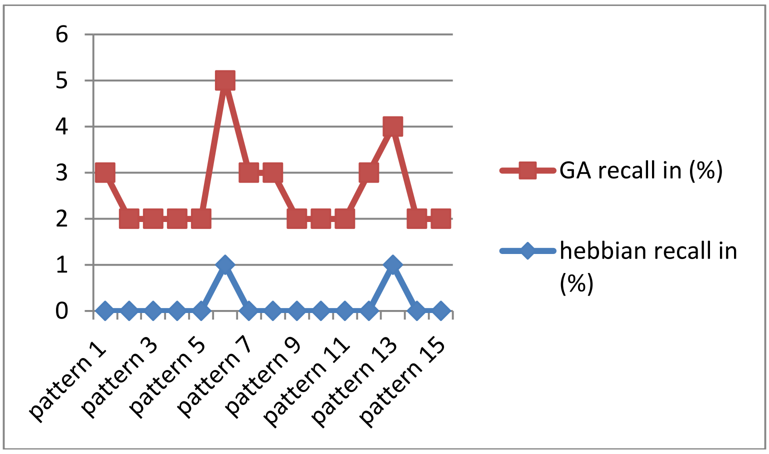
Fig 2: Graph representing the recalling accuracy of patterns in (%) where there is 1-bit noise induction



No. of patterns

Table 3: Results of recalling of patterns when there 2 bits errors in the input patterns

No. of PATTERNS	Recalling success in (%)	
	Hebbian rule	**GA**
PATTERN 1	0	3
PATTERN 2	0	2
PATTERN 3	0	1
PATTERN 4	0	2
PATTERN 5	0	2
PATTERN 6	1	4
PATTERN 7	0	3
PATTERN 8	0	3
PATTERN 9	0	2
PATTERN 10	0	2
PATTERN 11	0	2
PATTERN 12	0	3
PATTERN 13	1	3
PATTERN 14	0	2
PATTERN 15	0	2



No. of patterns

Fig 3: Graph representing the recalling accuracy of patterns in (%) where there is 2-bit noise induction

CONCLUSION

In this thesis work, from the above results, we concluded that the results obtained from genetic algorithm are far better than that of the Hebbian rule. For the taken set of input patterns containing 0, 1 and 2 bits errors, the recalling success rate of is genetic algorithm is much more than the Hebbian rule for the stored patterns in Hopfield neural network. The performance of GA is less than what was expected to be. One most common reason for this deviation in the results of GA may be the position(s) of bits reverted for the induction of noise in the recalling pattern.

For recalling the exact pattern on presentation of any prototype input pattern of already stored pattern, it is possible to obtain more than one weight matrices as optimal weight matrices from the generated set of weight matrices. The aim of this experiment is to provide an alternative approach for solving a pattern association problem. This concept for pattern recognition can be applied in different cases like for alphabets, objects, shapes, overlapped alphabets, etc.

REFERENCES

1. Ritter, G.X &Sussner, P 1996, 'An introduction to morphological neural networks', *Proc. 13thInt. Conf. Pattern recognition*, 4, pp. 709-717.
2. Mangal, M. & Singh, M.P 2007, 'Analysis of Multidimensional XOR Classification Problem with Evolutionary Feed Forward Neural Networks', *International Journal on Artificial Intelligence Tools*, 16(1), pp. 111-120.
3. Jabin, S 2010, 'Learning Classifier Systems Approach for Automated Discovery of Hierarchical Censored Production Rules In Proceedings of information and Communication Technologies', *Springer Berlin Heidelberg*, pp. 68-77.

4. Mangal, M & Singh, M.P 2006, 'Handwritten English Vowels using Hybrid Evolutionary Feed -forward Neural Network', *Malaysian Journal of Computer Science*, 19(2), pp. 169-187.
5. Singh, T.P &Jabin, S & Singh, M 2010, 'Evolving weight matrices to increase the capacity of Hopfield neural network associative memory using hybrid evolutionary algorithms', *In Proc. IEEE Int. Conf. Computational Intelligence & Computing Research(ICCIC)*, pp. 1-5.
6. Kodjabachain, J & Meyer, J.A 1995, 'Evolution and Development of Control Architectures in Animates', *Robot Autonomous Systems*, vol. 16, pp. 161-182.
7. Imada, A & Araki K 1997, ' Hopfield Model of Associative Memory as a Test Function of Evolutionary Computations', *The 1st International Workshop on Frontier in Evolutionary Algorithm, Proceedings of Joint Conference of Computer Science*, vol. 1, pp. 180-183.
8. Imada, A & Araki K 1997, 'Evolved Asymmetry and Dilution of Random Synaptic Weights in Hopfield Network Turn a Spin-glass Phase into Associative memory', *The 2nd International Conference on Computational Intelligence and Neuroscience proceeding of Joint Conference of Computer Science,* vol. 2, pp. 223-226.
9. Mangal M & Singh M.P 2007, 'Analysis of Classification for the Multidimensional Parity-Bit-Checking Problem with Hybrid Evolutionary Feed-forward Neural Network', *Neurocomputing*, vol. 70, pp. 1511-1524.
10. Burges, C.J.S 1995, 'Handwritten digit string recognition', *In M.A. Arbid, (eds): The Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press,* pp. 447-450.
11. Mori, S 1994, 'Historical review of theory and practice of handwritten character recognition, Fundamentals in Handwriting Recognition', *S. Impedovo (ed.), NATO ASI Series F Computer and Systems Sciences, Springer-Verlag*, vol. 124, pp. 43-69.
12. Hopfield, J.J 1982, 'Neural networks and physical systems with emergent collective computational abilities', *Proceedings of the National Academy Sciences, USA*, vol. 79, pp. 2554-2558.
13. Hebb, D 1949, 'The Organization of Behaviour', *A Neuropsychological Theory, Wiley, New York*.
14. Pal, S.K., De S &Ghosh 1997, 'Designing Hopfield type networks using genetic algorithms and its comparison with simulated annealing', *Intl Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, pp. 447-461.
15. He, J., Xu, J & Yao, X 2000, 'Solving Equations by Hybrid Evolutionary Computation Techniques', *IEEE Transactions on Evolutionary Computation*, vol. 4, No. 3.
16. Yao, X &Xu, Y 2006, 'Recent Advances in Evolutionary Computation', *J. Comput. Sci. & Tech Vol.*, vol. 21, pp. 1-18.
17. Islam, M. M & Yao, X 2003, 'Genetic Algorithm for Training Cooperative Neural Network Ensembles', *IEEE Transactions on Neural Networks*, vol. 14, pp. 4.
18. Yao, X 1999, 'Evolving artificial neural networks', *Proceeding of the IEEE*, vol. 87, pp. 1423-1447.
19. Imada, A & Araki K 1995, 'Genetic Algorithm Enlarges the Capacity of Associative Memory', *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 413-420
20. Ubiquity, Nov. 12-18 2013, vol. 4, Issue 37. http://www.acm.org/ubiquity
21. Freeman J.A &Skapura D.M 1991, 'Neural Network: Algorithms', *Applications and Programming Techniques, Reading, Ma: Assis on Wesley.*
22. Antognetti P &Milutinovic V, 'Neural Network: Concepts, Application, and Implementations9Eds', *Prentica Hall, Englewood Cliffs, NJ,* vol. I-IV, pp. 191.
23. Anderson, J.A 1988, 'Rosenfield E.: Neurocomputing: Foundations of Research MIT Press, Boston', MA.
24. Lee, D.L Mar. 2002, 'Pattern sequence recognition using a time-varying Hopfield network', *IEEE Trans. Neural Network,* vol. 13, no. 2, pp. 330-342.
25. Shen, D & Cruz Jr., J.B 2005, 'Encoding strategy for maximum noise tolerance', *IEEE Trans. Neural Networks,* 16, pp. 293–300.
26. Jain, A K., Robert, P W D & Mao, J 2002, 'Statistical Pattern Recognition: A Review', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, number 1, pp. 4-33.
27. Shapiro, J L 2001, 'Theoretical aspects of evolutionary computing', *Statistical Mechanics Theory of Genetic Algorithms, Natural Computing (Springer-Verlag, London, UK)*, pp. 87-108.
28. Kumar, S. & Singh, M. P 2010, 'Pattern Recall Analysis of the Hopfield Neural Network with a Genetic Algorithm', *Computers and Mathematics with Applications,* vol. 60(4), pp. 1049-1057.
29. Singh, T P., Singh, M P & Kumar, S 2010, 'Performance analysis of Hopfield model of a neural network with an evolutionary approach for pattern recalling', *International Journal of Engineering Science and Technology*, Vol. 2(4), pp. 504-511.
30. Salcedo-Sanz, S. & Yao, X. 'A Hybrid Hopfield Network-Genetic Algorithm Approach for the Terminal Assignment Problem', *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics,* vol. 34, No. 6, pp. 2343-2353.
31. Chartier, S &Proulx, R Nov. 2005, 'NDRAM: a Nonlinear dynamic recurrent associative memory for learning bipolar and non-bipolar correlated patterns', *IEEE Trans. Neural Network,* vol. 16, no. 6, pp. 1393–1400.
32. Sahoo, R.C & Kumar, S May 2016, 'Implementation of Hopfield Neural Network for its capacity with Finger Print Images', *International Journal of Computer Applications (0975 – 8887),* vol. 141, No. 5.
33. Wang, S & Wang, H Feb. 2008, 'Password authentication using Hopfield Neural Network', *IEEE Trans. On Systems, Man and Cybernetics Society*, vol. 38, pp. 265-268.
34. Shima, T., Ikuta, C., Uwate, Y &Nishio, Y Oct. 2011, 'Investigation of Recall image by Partitioned HNN', *IEEE Circuit Theory and Design (ECCTD*).
35. Singh, T.P &Jabin, S May 2012, 'Evolving connection weights for pattern storage and recall in Hopfield model of feedback NN using a GA', *International Journal on Soft Computing (IJSC),* vol. 3, No. 2.

36. Scandal, T 2015, 'Comparison of a different learning algorithm for PR with HNN', *2015 Annual International Conference on Biologically Inspired Cognitive Architectures*, vol. 71, pp. 68–75.
37. Bansal, S & Dixit, R 2016, 'Performance enhancement of the pattern recalling efficiency of Hopfield neural network using a genetic algorithm for cursive handwritten character recognition', *International Journal of Applied Pattern recognition,* vol. 3, issue 1.
38. Simon, D. 2013, 'Evolutionary Optimization Algorithms: Biologically-Inspired and Population-Based Approaches to Computer Intelligence', Hoboken: Wiley.

