



## Accurate Derivation and Efficient Implementation of Higher Order Logic Programming in Hopfield Network



SAGIR Abdu Masanawa<sup>1\*</sup>, IBRAHIM Sani Doro<sup>2</sup> & BAKO Deborah<sup>3</sup>

<sup>1</sup>Department of Mathematics, Federal University Dutsin-Ma, P.M.B. 5001 Dutsin-Ma, Katsina State. Nigeria.

<sup>2</sup>Mathematics Unit, School of General Studies, Federal University of Transportation, Daura. Katsina State. Nigeria.

<sup>3</sup>Department of Mathematics, Federal University of Technology Minna. Niger State. Nigeria.

\*Corresponding Author Email: [amsagir@fudutsinma.edu.ng](mailto:amsagir@fudutsinma.edu.ng)

### ABSTRACT

The complexity that arose naturally is how symbolic knowledge can be represented and dealt effectively within artificial neural networks. This paper describes the implementation of higher order logic programming in Hopfield network by carrying out computer simulation using NetLogo and running the relaxation for several trials and combination of neurons. In Hopfield network, optimization of logical inconsistency is carried out by the network after the connection strengths are defined from the logic program; the network relaxes to neural states which are models (i.e. viable logical interpretations) for the corresponding logic program. A solution of an optimization problem is obtained after the network is relaxed to an equilibrium state. Applications that benefitted on Hopfield network include solving NP-complete optimization problem such as travelling salesman problem; non-linear and discontinuous data in larger field and connections, and it is able to detect all possible interactions between predictor variables such as detect complex nonlinear relationships of dependent and independent variables.

### Keywords:

Artificial neural Network,  
Agent-based modelling,  
Hopfield network,  
Logic programming

### INTRODUCTION

Artificial neural network (ANN) is a simplified mathematical model of the human brain. It is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. It can be implemented by both electric elements and computer software. It is a parallel distributed processor with large numbers of connections; it is an information processing system that has certain performance characters in common with biological neural networks (Galushkin, 2007). The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems.

ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well. ANN is a system loosely modeled on the human brain. The field goes by many names, such as connectionism; parallel distributed processing, neurocomputing,

natural intelligent systems, machine learning algorithms, and ANNs. It is an attempt to simulate within specialized hardware or sophisticated software, the multiple layers of simple processing elements called neurons. Each neuron is linked to certain of its neighbors with varying coefficients of connectivity that represent the strengths of these connections (Pranob, Charles, Rajesh-Kumar, Nikhita, Santhosh, Harish and Swathi, 2012).

Hopfield network is a feedback network otherwise called recurrent network. Hopfield network and Logic programming are two important aspects in artificial neural network. In the beginning of the 1980,s; Hopfield published two scientific papers which attracted interest. He presented a model which consisted of a set of first – order (nonlinear) differential equations that minimize a certain “energy” function (Hopfield, 1982). Hopfield argued that there are many significant computational capabilities at the network level which are nonexistent at the neural level (Hopfield, 1982; Hopfield, 1984). This was the starting point of the new era of neural networks, which continues today. Hopfield showed that models of physical systems could be used to solve computational problems (Andrey, 2011).

Hopfield network is a recurrent network having synaptic connection pattern such that there is an underlying Lyapunov function for the activity dynamics. Started in any initial state, the state of the system evolves to a final state that is a (local) minimum of the Lyapunov function (Hopfield, 1984).

On the other hand, Logic programming is a programming paradigm based on formal logic in which a program is written as a set of sentences in logical form; expressing fact and rules about problem domains (Saratha, 2006). Logic programming applies to all areas of artificial intelligence and computer science and is fundamental to all of them. It is applicable in logic programming methodologies, inductive logic programming, multi-relational data mining, knowledge representation, non-monotonic reasoning, semantic web reasoning, databases, implementations and architectures and constraint logic programming (Michael, Andreas, Ramakrishnan and Ulrich, 2001).

Conjunction normal form (CNF) is widely used in an area of logical reasoning systems. In order to find models, a set of Horn clauses which are logical clauses of the form  $A \leftarrow B_1, B_2, \dots, B_n$  would be implemented in Saratha's method for calculation the synaptic strengths, where the arrow is read as "if" and the commas "and", is given and the aim is to find 'models' corresponding to the given logic program. The model here refers to a setoff interpretation which satisfied the logical clauses (Saratha, 2009a).

The general framework of this research work is focused on integration logic programming in Hopfield network based on Saratha's method (Saratha, 2010a); and by carrying out computer simulations using NetLogo, so as to enable us obtain good solutions with less complexity.

Artificial neural networks can be most adequately characterized as 'computational models', whose operations are based on parallel processing and have particularly the following properties:

- (I) The ability to adapt or learn
- (II) The recall
- (III) The ability to generalize or to cluster or organize data from training pattern.

Research on Neural-symbolic integration has made significant progress in the recent years. Neural-symbolic integration has led to a number of systems developments in the 1980's and 1990's. Some researchers have written on comprehensive papers on artificial neural networks such as in: Neural-symbolic learning systems, Rule learning by searching on adapted networks and Connectionist inference models. Saratha (2010a) and other authors contributed on logic programming in neural networks.

Saratha and Wan Abdullah (2008) showed that Hebbian learning for governing events corresponding to some respective program clauses is equivalent with learning using Wan Abdullah (1996) for the same respective program clauses. This has been shown experimentally through computer simulations. They presented doing logic programming in Hopfield network. A set of Horn clauses which are logic clauses of the form  $A \leftarrow B_1, B_2, \dots, B_n$  was used. The aim here was to find the set(s) of interpretation i.e. the truth value for the atoms in the clauses which yield all the clauses true. This enable them find models corresponding to the given logic program. They consider the following logic program  $A \leftarrow B, C, D \leftarrow B, C \leftarrow$  whose three clauses translate respectively as  $A \rightarrow B \rightarrow C, D \rightarrow B$  and  $C$ . The values 1 to true and -1 to false were assigned and  $\neg P = -1$  and  $\neg P = 1$  indicates a consistent and unsatisfied respectively. Therefore, looking for a consistent interpretation is a combinatorial minimization of the inconsistency, the value of  $\neg P$ . Hopcroft and Ullmann (1979), described for clauses involving more than three atoms (spins) can be reduced to conjunction normal form of atomistic clauses  $C_i$  containing only three atoms (spins).

Saratha (2010b), outline the doing logic programming on a Hopfield network as in (Hopcroft and Ullmann 1979). She further described the equalities between the Conjunction Normal Form (CNF) and Disjunction Normal Form (DNF) conversion in carrying out logic program in Hopfield network as:

- (I) CNF is much more commonly in the area of logical reasoning systems, while DNF is very commonly used in circuit design.
- (II) In CNF, every expression is a conjunction of disjunctions of literals, i.e.  $F = F_1, F_2, \dots, F_n, n \geq 1, F_1$  is a disjunction of literals and  $F = F_1, F_2, \dots, F_n, n \geq 1$  is its conjunct. While in DNF, every expression is a disjunction of conjunctions of literals i.e.  $F = F_1, F_2, \dots, F_n, n \geq 1, F_1$  is a conjunction of literals and  $F = F_1, F_2, \dots, F_n, n \geq 1$  is its disjuncts.
- (III) In Boolean logic, the number of models for CNF is higher and reliable than the DNF conversion.

She proved the results by comparing the truth tables of CNF and DNF conversion, that DNF is not as good as CNF, despite the fact that CNF is still stable when the number of literals per clause increased also the number of

unsatisfied events increased. CNF has the advantage of it is its uniformly formed form, which makes it suitable to automatic processing which needs to define the rule for the machine to recognize the logic.

Following learning rules comparison in Neuro-symbolic integration, Saratha (2011) analyzed the performance comparison of doing logic programming in Hopfield network by using different learning rules such include Hebbian learning rule which is local and incremental, but

has a low absolute storage capacity of  $\frac{n}{2 \ln(n)}$  overall,

its performance is poor in terms of storage capacity, attraction, and spurious memories. Three different learning rules were employed such as Pseudo-inverse learning rule (PI) also known as Projection learning rule and Hyperbolic Activation Function otherwise known as sigmoid function. The sigmoid function type of activation function puts much emphasis on minor noise perturbation instead of the signals related to the cost and constrains encoded in the network. This function, however, can tolerate with noise and do perform well when the network gets larger.

In order to ascertain the new rule for energy relaxation, Saratha (2009b) has presented a paper energy relaxation for Hopfield network with the new learning rule. A method of doing logic programming on a Hopfield network has been described as in (Hopcroft and Ullmann 1979). The new learning rule introduced by Stokey (1997), it was defined as the weight matrix  $w_{ij}$  of an attractor neural network is said to follow the new learning rule if it obeys  $w_{ij}^0 = 0 \forall i, j$  and

$$w_{ij}^v = w_{ij}^{v-1} + \frac{1}{n} \xi_i^v \xi_j^v - \frac{1}{n} \xi_i^v h_{ji}^v - \frac{1}{n} h_{ji}^v \xi_j^v \quad \text{where}$$

$$h_{ij}^\mu = \sum_{k=1, k \neq i, j}^n w_{ik}^{\mu-1} \xi_k^\mu \text{ is a form of local field at neuron I}$$

(the input to the neuron i), and  $\xi^\mu$  is the new pattern to be learnt ( $\xi_i^\mu = \pm 1$ ). The energy landscape and fitness evaluation has been carried out. Results have shown that the energy landscapes formed by the network programmed by program clauses are rather flat due to the zero fitness values. Comparison between Hebbian and the new method revealed that the new rule performs better as it has capacity and does not suffer significant capacity loss when input clauses get larger.

Saratha (2010c) reviewed a method of doing logic program on a Hopfield network based on energy minimization scheme but restricted to program clauses of up to three atoms. They pointed out two phases to the operation of the discrete Hopfield network as content – addressable memory, namely the storage phase and the

retrieval phase. The problem is to determine whether there exists a truth assignment to variables appearing in a Boolean formula  $\Phi$  in conjunction normal form such that  $\Phi$  is satisfied (true). They stated that, a formula F is said to be satisfiable if and only if there exist an interpretation, I such that I models F. Fitness landscape and Negation as failure have been discussed. In neural network fitness landscapes correspond to energy landscapes.

Xinchua and Tony (1999) show that the performance of the Hopfield network can be improved by using a relaxation rate to control the relaxation process, in order to achieve solutions with better quality. Two types of relaxation rate (constant and dynamic) are proposed and evaluated through simulations based on 200 randomly generated city distributions of the 10-city traveling salesman problem. The result shows that using a relaxation rate can decrease the error rate by 9.8% and increase the percentage of valid tours by 14.0% as compared to those without using a relaxation rate. Using a dynamic relaxation rate can further decrease the error rate by 4.2% and increase the percentage of valid tours by 0.4% as compared to those using a constant relaxation rate. Some researchers such as Abubakar, Sagir, Yusuf & Boaku (2021); Abubakar, Yusuf & Sagir (2020); Abubakar, Sagir, Yusuf & Abdurrahman (2020); Abubakar, Sabri, Sagir & Yusuf (2020); Abubakar, Sagir & Yusuf (2020) have written on Hopfield network.

## MATERIALS AND METHODS

### Derivation of the Method

Consider Saratha (2010a) for the derivation and implementation of higher order logic programming in Hopfield network:

$$\eta = (P \leftarrow Q, R)(S \leftarrow Q, T)(R \leftarrow) \quad (1)$$

Given the goal  $\leftarrow \varphi$ . That is  $\eta \wedge \neg \varphi$  was considered as inconsistent in order to prove the goal. Alternatively, an interpretation for the Herbrand base of the problem which is consistent with  $\eta$  (which yields  $\eta$  true) and examine the truth of  $\xi$  in such an interpretation was required.

### Algorithm of Logic Programming in Hopfield Network

The following algorithms show on how a logic program can be done in Hopfield network:

(i) All the given clauses in the logic program are translated into Boolean algebraic form

$$P = (A \neg B \neg C)(D \neg B) C \quad (2)$$

$$\neg P = (\neg ABC)(\neg DB)(\neg C) \quad (3)$$

From the above equations (2) & (3), bipolar neurons which minimized a cost function were adopted, when all the clauses are satisfied. However, from (3.3) it shows:

$$E_p = \frac{1}{2}(1 - S_A) \frac{1}{2}(1 + S_B) \frac{1}{2}(1 + S_C) + \frac{1}{2}(1 - S_D) \frac{1}{2}(1 + S_B) + \frac{1}{2}(1 - S_C)$$

$$E_p = \frac{1}{2}(1 - S_A) \frac{1}{2}(1 + S_B) \frac{1}{2}(1 + S_C) + \frac{1}{2}(1 - S_D) \frac{1}{2}(1 + S_B) + \frac{1}{2}(1 - S_C) \tag{4}$$

$E_p = 0$  (1 if  $S_x$  is true, and  $-1$  if  $S_x$  is false), i.e.  $E_p = 0$  when all clauses are satisfied.

From (4), the following was obtained as:

$E_p(0) = \frac{7}{8}(S_A = S_B = S_C = S_D = 0)$ . In this instance  $E$  has minimum energy known as global minimum energy value, i.e.

$$E_{\min} = E_p - E_p(0) = 0 - \frac{7}{8} = -\frac{7}{8}$$

Also, It has been proven from the truth table for  $P = (A \leftarrow B, C, D \leftarrow B, C \leftarrow)$  that if  $E_p = 0$ , this implies that  $E_p \propto$  No. of clauses unsatisfied.

(ii) An energy function:

$$H = -\frac{1}{3} \sum_A \sum_B \sum_C J_{(ABC)}^{(3)} S_A S_B S_C - \frac{1}{2} \sum_A \sum_B J_{(AB)}^{(2)} S_A S_B - \sum_A J_A^{(1)} S_A$$

An energy function been defined as (5), where the synaptic strength is completely symmetric  $w_{ijk} = w_{kji}, \forall_{ijk}$  with zeros in the diagonal planes i.e.  $w_{iii} = 0, \forall_i$

Consider permutation:

$$P(n, r) = nP_r = \frac{n!}{(n-r)!}, \text{ i.e.}$$

$$P(4, 3) = 4P_3 = \frac{4!}{(4-3)!} = 24$$

Hopcroft and Ullmann (1979), for clauses involving more than three atoms (spins) can be reduced to conjunction normal form of atomistic clauses  $C_i$  containing only three atoms (spins).

(iii) The updating rule reads:

$$S_i(t+1) = \text{sgn}[h_i(t)] \tag{6}$$

where local field is given by the given equation below:

$$h_i(t) = -\frac{1}{3} \sum_A \sum_B \sum_C J_{(ABC)}^{(3)} S_A(t) S_B(t) S_C(t) - \frac{1}{2} \sum_A \sum_B J_{(AB)}^{(2)} S_A(t) S_B - \sum_A J_A^{(1)} S_A(t) \tag{7}$$

The cost function  $E_p$  and the energy function H, have been compared, which yields the value of synaptic strengths as shown below:

For instance, the values of  $J_{(ABC)}^{(3)}, J_{(AB)}^{(2)}$  e.t.c. were obtained by expanding and considering the coefficients:

$$J_{(ABC)}^{(3)} = \lambda^3 \left( -\frac{1}{8} S_A S_B S_C \right) = \frac{1}{(3-1)!} \left( -\frac{1}{8} (+1)(+1)(+1) \right) = -\frac{1}{16} = \frac{1}{16}$$

(Reverse Analysis)

$$J_{(ABD)}^{(3)} = 0, J_{(ACD)}^{(3)} = 0, J_{(BCD)}^{(3)} = 0$$

$$J_{(AB)}^{(2)} = \lambda^2 \left( -\frac{1}{8} S_A S_B \right) =$$

$$\frac{1}{(2-1)!} \left( -\frac{1}{8} (+1)(+1) \right) = -\frac{1}{8} = \frac{1}{8}$$

(Reverse Analysis); e.t.c.

(5) The Hebbian learning rule for a two - neuron synaptic connection was obtained as:

$$J_{(i,j)}^{(2)} = \lambda_2 S_i S_j \text{ --- } S_m \tag{8}$$

where  $\lambda_2$  is a learning rate.

For connection of the order  $n$ , between  $n$  neurons,  $S_i, S_j, \text{ ---}, S_m$  Equation (9) was generalized, this to:

$$J_{(i,j, \text{ --- } m)}^{(n)} = \lambda_n S_i S_j \text{ --- } S_m \tag{9}$$

This gives the changes in synaptic strengths depending on the activities of the neurons. Thus, the Hebbian learning as in (8) & (9) above corresponds to hardwiring the neural network with synaptic strengths obtained using Saratha

(2010a), provided that  $\lambda_n = \frac{1}{(n-1)!}$  is true.

By translating all clauses and negation of it into Boolean algebraic form, the proposed truth table could be translated as the Truth tables of rows  $2^n$ , where  $n = 5$ , see Table 1.

Table 1: The truth table for the number of unsatisfied clauses and  $E_\eta$  for

$$\eta = (P \leftarrow Q, R)(S \leftarrow Q, T)(R \leftarrow)$$

$\xi_P$	$\xi_Q$	$\xi_R$	$\xi_S$	$\xi_T$	$P \leftarrow Q, R$	$S \leftarrow Q, T$	$R \leftarrow$	No. UnS. Clauses	$E_\eta$
-1	-1	-1	-1	-1	1	1	-1	1	1
-1	-1	-1	-1	1	1	1	-1	1	1
-1	-1	-1	1	-1	1	1	-1	1	1
-1	-1	-1	1	1	1	1	-1	1	1
-1	-1	1	-1	-1	1	1	1	0	0
-1	-1	1	-1	1	1	1	1	0	0
-1	-1	1	1	-1	1	1	1	0	0
-1	-1	1	1	1	1	1	1	0	0
-1	1	-1	-1	-1	1	1	-1	1	1
-1	1	-1	-1	1	1	-1	-1	2	2
-1	1	-1	1	-1	1	1	-1	1	1
-1	1	-1	1	1	1	-1	-1	2	2
-1	1	1	-1	-1	-1	1	1	1	1
-1	1	1	-1	1	-1	-1	1	2	2
-1	1	1	1	-1	-1	1	1	1	1
-1	1	1	1	1	-1	-1	1	2	2
1	-1	-1	-1	-1	1	1	-1	1	1
1	-1	-1	-1	1	1	1	-1	1	1
1	-1	-1	1	-1	1	1	-1	1	1
1	-1	-1	1	1	1	1	-1	1	1
1	-1	1	-1	-1	1	1	1	0	0
1	-1	1	-1	1	1	1	1	0	0
1	-1	1	1	-1	1	1	1	0	0
1	-1	1	1	1	1	1	1	0	0
1	1	-1	-1	-1	1	1	-1	1	1
1	1	-1	-1	1	1	1	-1	1	1
1	1	-1	1	-1	1	1	-1	1	1
1	1	-1	1	1	1	1	-1	1	1
1	1	1	-1	-1	1	1	1	0	0
1	1	1	-1	1	1	1	1	0	0
1	1	1	1	-1	1	1	1	0	0
1	1	1	1	1	1	1	1	0	0

The synaptic strengths, as well as the cost function for bipolar neurons, which is minimized when all the clauses are satisfied are presented in the following equations:

$$\eta = (P \neg Q \neg R)(S \neg Q \neg T)(R) \tag{10}$$

$$\neg \eta = (\neg PQR)(\neg SQT)(\neg R) \tag{11}$$

$$E_\eta = \frac{1}{2}(1 - P) \frac{1}{2}(1 + Q) \frac{1}{2}(1 + R) + \tag{12}$$

$$\frac{1}{2}(1 - S) \frac{1}{2}(1 + Q) \frac{1}{2}(1 + T) + \frac{1}{2}(1 - R)$$

**Implementation Analysis**

In the implementation analysis, computer simulations been carried out using NetLogo (Wilensky, 2016). NetLogo was used to design agent based modeling (ABM) in Netlogo by using button, input, output, slides and other functions that make ABM easy to understand and use. In this simulator, programmer needs to design an agent based modeling to implement the higher order Hopfield network. Agent-Based Modelling (ABM), which also called individual-based modelling is a new computational modelling paradigm which is an analysing systems that representing the ‘agents’ that involving and simulating of their interactions (Grimm, Volker; Railsback and Steven, 2005). Their attributes and

behaviours will be group together through their interactions to become a scale.

**RESULTS AND DISCUSSION**

The relaxation was run for 1000 trial and 100 combinations of neurons so as to reduce statistical error. The selected tolerance value is 0.001. All these values are

obtained by trial and error technique. The solutions obtained are global solutions, so the distance between the stable states and the attractors are zero. This shows zero value was obtained for Hamming distance, which indicates that the stable states for both learning are the same. The clauses and corresponding synaptic strengths were obtained.

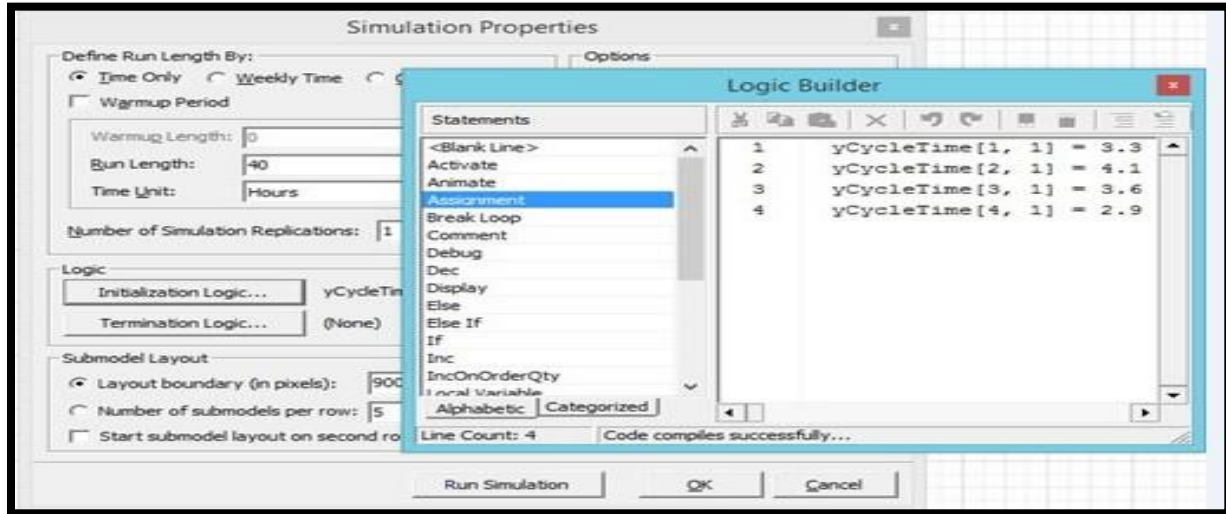


Figure 1. Simulation Properties

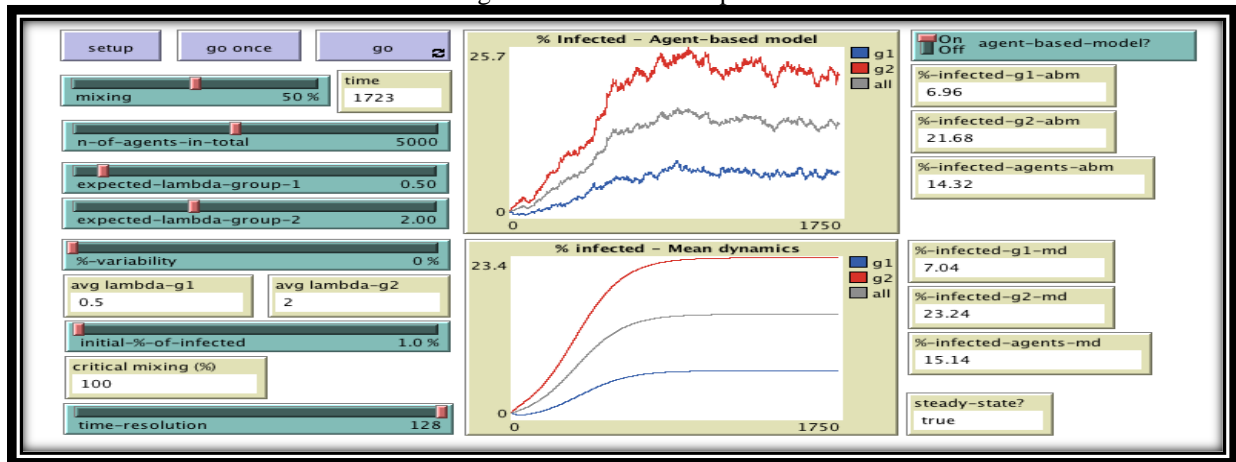


Figure 2. Global Solution for Hamming Distance

**Traveling Salesman Problem**

Applications that benefitted on Hopfield network include solving NP-complete optimization problem such as travelling salesman problem; non-linear and discontinuous data in larger field and connections (Potvin, J-Y, 1995; Dantzig, Ray and Johnson, 2011). Thus, there is no known algorithm that will solve it in polynomial time. The main use is to find the shortest closed path from among a group of cities. It is also used to find the cheapest closed distance. The total number of the tours was reduced to those that contain the short tours.

The input data in the network is the distance between each pair of cities. The target output in the network is the satisfactory closed tours' distance. Accordingly, there are  $(N - 1)!$  different closed tours between N cites. TSP asks for the cheapest possible tour through a given collection of cities.

**Problem 1:** Consider the following distances between six distinct cities, as presented in Table 2.

Table 2: Distances between six cities (in Km)

Cities Number	1	2	3	4	5	6
1	0	250	400	200	500	200
2	250	0	350	330	250	200
3	400	350	0	550	300	200
4	200	330	500	0	300	200
5	500	250	300	200	0	200
6	200	200	200	200	200	0

Table 2 can be presented in matrix form as:

$$\begin{bmatrix} 0 & 25 & 40 & 20 & 50 & 20 \\ 25 & 0 & 35 & 33 & 25 & 20 \\ 40 & 35 & 0 & 55 & 30 & 20 \\ 20 & 33 & 50 & 0 & 30 & 20 \\ 50 & 25 & 30 & 20 & 0 & 20 \\ 20 & 20 & 20 & 20 & 20 & 0 \end{bmatrix} \times 10\text{km} \quad (13)$$

This leads to equation (14) as:

$$x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4 \rightarrow x_5 \rightarrow x_6 \rightarrow x_1 \quad (14)$$

where each  $x_i, i = 1, \dots, 6$  is a city in a tour

The training data set is chosen to be between 5 and 500. After simulation, the following results in Table 3 and Figure 3 were obtained.

Table 3. Distance between 5 Cities

Cities Number	1	2	3	4	5
1	0	1	100	100	10
2	1	0	300	100	1
3	100	100	1	1	200
4	100	300	0	0	1
5	10	1	1	1	0

Thus, by taking the satisfactory values between 100 and 1000, the best tour was found to be:

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 3 \rightarrow 5$$

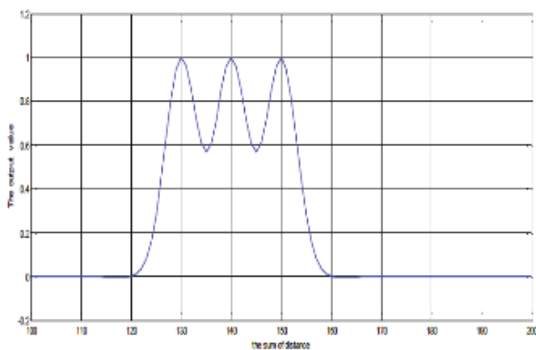


Figure 3: The output curve which represents Table 2

### CONCLUSION

In this research work the derivation and implementation of higher order logic programming in Hopfield network were carried out. This provides the means to find a set of interpretation for the atoms in the clauses which yield all the clauses true. The energy relaxation process of Hopfield network in doing logic programming been analyzed. It was proven that the relaxation has an important impact on the performance of the network. It is used to determine whether there exists a truth assignment to variables appearing in a Boolean formula in conjunction normal form such that the formula is satisfied (true). Applications that benefitted on Hopfield network include solving NP-complete optimization problem such as travelling salesman problem was demonstrated.

### REFERENCE

Abubakar, H., Masanawa, A. S., Yusuf, S., & Boaku, G. I. (2021). Optimal representation to high order random boolean ksatisfiability via election algorithm as heuristic search approach in hopeld neural networks. *Journal of the Nigerian Society of Physical Sciences*, 201-208. <https://doi.org/10.46481/jnsps.2021.217>

Abubakar, H., Masanawa, S. A., & Yusuf, S. (2020). Neuro-symbolic Integration of Hopfield Neural Network for Optimal Maximum Random ksatisfiability (maxrksat) Representation. *Journal of Reliability and Statistical Studies*, 199-220. <https://doi.org/10.13052/jrss1550-4646.13110>

Abubakar, H., Yusuf, S., & Masanawa, S. A. (2020). Exploring the Feasibility of Integrating Random k-Satisfiability in Hopfield Neural Network. *International Journal of Modern Mathematical Sciences*, 18, 92. <https://doi.org/10.3390/pr8050568>

Abubakar, H., Masanawa, S. A., Yusuf, S., & Abdurrahman, Y. (2020). Agent Based Computational Modelling For Mapping Of Exact Ksatisfiability Representation In Hopfield Neural Network Model. *Int. J. Sci. tech. Res*, 9(08), 7685. <https://doi.org/10.13052/jrss1550-4646.13110>

Abubakar, H., Sabri, S. R. M., Masanawa, S. A., & Yusuf, S. (2020). Modified Election Algorithm in Hopfield Neural Network for Optimal Random k Satisfiability Representation. *International Journal for Simulation and Multidisciplinary Design Optimization*, 11, 16. <https://doi.org/10.3390/pr8050568>

Andrey P. (2011). Hopfield Neural Network. Retrieved from <https://andreypudov.com/p/hopfield-neural-network.html>.

- Dantzig, G., Ray, F. and Johnson, S. (2011). Solution of Large Scale Traveling Salesman Problem. *Journal of the Operations Research Society of America*. 2(4): 393–410.
- Galushkin, A.T. (2007). *Neural Networks Theory*. Springer, Berlin, Germany.
- Grimm, Volker; Railsback, Steven F. (2005). *Individual-Based Modeling and Ecology*. Princeton University Press, New Jersey.
- Hopcroft, J.E., and Ullmann, J.D. (1979). *Introduction to Automata Theory, Language and Computation*. Addison – Wesley Publication, Pearson Education, India.
- Hopfield, J.J. (1982). Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of National Academy of Sciences*. U.S.A. 79(8): 2554 – 2558.
- Hopfield, J.J. (1984). Neurons with Graded Response Have Collective Computational Properties Like Those of Two – State Neurons. *Proceedings of National Academy of Scienc*. U.S.A. 81(10): 3088 – 3092.
- Michael, M., Andreas, P., Ramakrishnan, C.R., and Ulrich, U. (2001). Theory and Practice of Logic Programming. 1(5): 631 – 632. doi: 10.1017/S1471068401001089. Online @ [http://journals.cambridge.org/abstract\\_S1471068401001089](http://journals.cambridge.org/abstract_S1471068401001089)
- Potvin, J-Y (1995). *The Traveling Salesman Problem: A Neural Network Perspective*. Universite de Montreal, Centre de Recherche sur les Transports. Pp. 4 – 15.
- Pranob, K., Charles, H.K., Rajesh-Kumar, C., Nikhita, N., Santhosh, R., Harish, V., and Swathi, M. (2012). Artificial Neural Network Based Image Compression using Levenberg – Marquardt Algorith. *International Journal of Modern Engineering Research (IJMER)*. 1(2): 482 – 489.
- Saratha, S. and Wan Abdullah, W.A.T. (2008). Logic Learning in Hopfield Networks. arXiv preprint arXiv: 0804.4075.
- Saratha, S. (2006). *Logic Mining in Neural Networks*. PhD Thesis, Unpublished, University of Malaya, Kuala Lumpur, Malaysia.
- Saratha, S. (2009a). Enhancing Logic Programming in Recurrent Hopfield Network. *European Journal of Scientific Research*. 37(3): 361 – 367.
- Saratha, S. (2009b). Energy Relaxation for Hopfield Network with the New Learning Rule. *International Conference on Power Control and Optimization*. Bali, Indonesia.
- Saratha, S. (2010a). Upgrading Logic Programming in Hopfield Network. *Journal of Sains Malaysia*. 39(1): 115 – 118. Online @ [www.ukm.my/jsm/pdf\\_files/SM-PDF-39-1-2010/17.pdf](http://www.ukm.my/jsm/pdf_files/SM-PDF-39-1-2010/17.pdf).
- Saratha, S. (2010b). Clauses Representation Comparison in Neuro – Symbolic Integration. *Proceedings of the World Congress on Engineering*. Vol. 1. London, U.K.
- Saratha, S. (2010c). The Satisfiability Aspect of Logic on Little Hopfield Network. *American Journal of Scientific Research*. 7: 90 – 105. © EuroJournals Publishing, Inc. Online @ <http://www.eurojournals.com/ajsr.htm>
- Saratha, S. (2011). Learning Rules Comparison in Neuro – Symbolic Integration. *International Journal of Applied Physics and Mathematics*. 1(2): 129 – 132.
- Stokey, A.J. (1997). Increasing the Capacity of the Hopfield Network without Sacrificing Functionality. In Gerstner W. & Germond A. and Hastler M. & Nicoud J. (Eds.), ICANN97: Lecture Notes in Computer Science 1327, pp. 451 – 456, Springer – Verlag.
- Wan Abdullah, W.A.T. (1996). Logic Programming in Neural Networks. *Malaysian Journal of Computer Science*. 9(1): 1 – 5.
- Wilensky U. (2016). NetLogo. Retrieved from <https://ccl.northwestern.edu/netlogo/>
- Xinchuan, Z. and Tony, R.M. (1999). Improving the Performance of the Hopfield Network by using Relaxation rate. *Proceedings of the International Conference on Neural Networks and Genetic Algorithms*.