International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-4, April 2020

Agent-based Modeling for Activation Function in Enhancement Logic Programming in Hopfield Neural Network

Shehab Abdulhabib Alzaeemi, Saratha Sathasivam, Muraly Velavan, Mustafa bin Mamat

Abstract*: *Artificial Neural Network (ANN) uses many activation functions to update the state on neuron. The research and engineering have been used activation functions in the artificial neural network as the transfer functions. The most common reasons for using this transfer function were its unit interval boundaries, the functions and quick computability of its derivative, and several useful mathematical properties in the approximation of theory realm. Aim of this study is to figure out the best robust activation functions to accelerate HornSAT logic in the Hopfield Neural Network's context. In this paper we had developed Agent-based Modelling (ABM) assessed the performance of the Zeng Martinez Activation Function (ZMAF) and the Hyperbolic Tangent Activation Function (HTAF) beside the Wan Abdullah method to do Logic Programming (LP) in Hopfield Neural Network (HNN). These assessments are carried out on the basis of hamming distance (HD), the global minima ratio (zM), and CPU time. NETLOGO 5.3.1 software has been used for developing Agent-based Modeling (ABM) to test the proposed comparison of the efficaecy of these two activation functions HTAF and ZMAF.

Keywords* : *Logic Program, Hopfield Neural Network, Zeng Martinez Activation Function, Hyperbolic Tangent Activations Function, Agent based Modelling.

I. INTRODUCTION

Artificial Neural Network (ANN) is an exciting field in research due to providing an alternative style of doing computation and ANN is a leap towards the understanding of intelligence artificial (AI) [1]. The common behavior of the neural networks then exhibits high-level behavior such as the ability to data recognition, learn, and recall [2]. Hopfield Neural Network (HNN) has many interesting applications, implementations and features such as content addressable memory and fault tolerance [3]. Learning in HNN is accomplished by modifying the strength of the connect between the neurons in the network and the parameter of the threshold. This field of learning in the neural network consists of bipolar status neurons that are correlated by the asymmetric network structure. The connections of these neurons in the network are weighted within the sign of the synaptic weights between the neurons, which can be activated by using the activation functions [4].

Wan Abdullah offered a method to do a HornSAT logic in Hopfield Neural Network (HNN) [1]. The method proposed is an optimization of logic inconsistency where the synaptic weights between the neurons are defined and calculated from the Logic Programming (LP). Synaptic weights between the input neurons and the output neurons are calculated by compare energy function with cost function. After that the network relaxes to its neural state which is the model corresponding to the logic.

In Hopfield Neural Network (HNN), a final solution of the optimization is obtained if the output of the network is updated efficiently. In order to embrace the network optimally, the activation function will be assimilated with the LP in order to update the state of neurons in HNN [5, 6]. This paper shows that the performance of the HNN can be improved by using various activation functions based on Agent-Based Modelling.

This paper used Agent-Based Modeling (ABM) to model HornSAT logic in Hopfield Neural Network (HNN) with different activation functions: Zeng Martinez Activation Function (ZMAF) and Hyperbolic Tangent Activation Function (HTAF). The purpose of this study is to develop ABM to accelerate the performance of doing logic programming in Hopfield network by using two activation functions called HTAF and ZMAF.

II. RESEARCH METHOD

II.1 Hopfield Neural Network (HNN)

HNN is one of neural network models most widely utilized. It has feedback connections to a single neural network model. HNN systemically stores patterns [15] as a content addressable memory (CAM). HNN is a N-connected neuron network where every neuron's output and input are connected. w_{ij} is the connection weight from neuron i to j. HNN is the symmetric networks as mean $w_{ij} = w_{ji}$, and no self-feedback connections as mean $w_{ii} = w_{jj} = 0$. The following equation showed the general updating rule in HNN [25]:

Revised Manuscript Received on April 21, 2020.
* Corresponding Author
Shehab Abdulhabib Alzaeemi1*, Saratha Sathasivam2,*, School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang Malaysia. Email: saratha@usm.my
Muraly Velavan, School of General and Foundation Studies, AIMST University, 08100 Semeling, Kedah.
Mustafa bin Mamat, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, 21300 Kuala Terengganu, Terengganu.

$$S_i = \begin{cases} 1 & if \ \ h_i(t) > \theta_i \\ -1 & Otherwise \end{cases} \quad (1)$$

where S_i is the status or outputs of the i^{th} unit, $h_i(t)$ is the local field of HNN, and θ_i is threshold of unit i. The following equation shows the local field of HNN:

$$h_i(t) = \sum_j J_{ijk}^{(3)} S_j S_k + \sum_j w_{ij}^{(2)} S_j + w_i^{(1)} \quad (2)$$

The updating rule of the neurons in HNN is given by equation:

$$S_i(t+1) = \mathrm{sgn}\left[h_i(t)\right] \quad (3)$$

Lyapunov or energy equation will be used to investigate the final state of the neurons in HNN:

$$E_P = -\frac{1}{3}\sum_i \sum_j \sum_k w_{ijk}^3 S_i S_j S_k - \frac{1}{2}\sum_i \sum_j w_{ij}^{(2)} S_i S_j - \sum_i w_i^{(1)} S_i \quad (4)$$

Equation (3) updating rule guarantees that the energy in the network is minimized. The degree of HNN neuronal convergence is critical for Equation (4) [16, 17]. The energy value is therefore critical for a minimum global and local solution to be segregated. The HNN model's learning process is established by integrating the HornSAT behavior into the HNN. The behavior of HornSAT LP in HNN can be generalised by getting correct synaptic weights between the neurons and active the status of the neurons by HTAF and ZMAF.

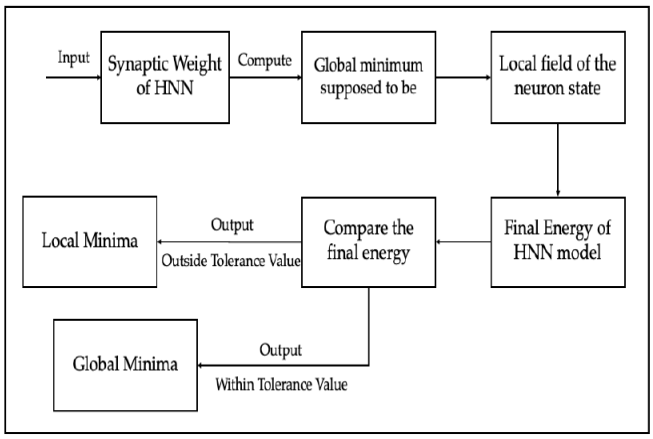

Figure 1. Summary of Hopfield Neural Network [18]

II.2 HornSAT Logic Programming

In this study, we would frequently encounter instances of HornSAT logic is a logic to deciding the satisfiability of sets of clauses. Horn clauses are clauses that have one or at most one positive literals. HornSAT Boolean variable can take the value of either -1 (false) or 1 (true). The followed components of HornSAT logic:

(a) It consists of a group n variables as $x_1, x_2 ..., x_n$

(b) A collection of literals in the clauses. A literal is a variable x or a negation of a variable $\overline{x}$.

(c) A group of m feature clauses: $L_1, L_2, ..., L_m$ . . Each clauses combined by just $\wedge$ logic AND. Every clause composed of literals joint by just $\vee$ logic OR.

The aim of HornSAT logical is to decide if there subsists an assignment of the true value of the variables in the clauses that make the P become satisfiability. HornSAT is a main impulsion in this paper as show in the following example of HornSAT logical programming,

$$\begin{aligned} P = A &\leftarrow B, C \\ \wedge D &\leftarrow B \\ \wedge C & \end{aligned} \quad (4)$$

The goal will be given

$$\leftarrow G$$

AV¬BV¬C, DV¬B and C are sequentially transmuted from (4) and G is a conjunction or the goal of neurons. The synaptic weight of the variables in the clauses of HornSAT logical programming P in the equation (4) can be determined by using Wan Abdullah's method [1].

II.3 Types of Activation Functions

There are a lot of researchers to used activation functions to active their net inputs as Bekir and Vehbi[3] proposed their network inputs utilizing a Scalar-to-Scalar function called the "Threshold Function or Transfer Function" and network effects called activation unit was the numerous significant units in the construction of the neural network. The results are the net outputs. A neuron output amplitude is limited by an activation function. The authors[3, 4] have proposed that artificial neural networks sum up the synaptic weights and input signal products and generate the output signal or activate the status of the neurons by an activation function as HTAF, sigmoid function, or ZMAF, etc. The most common activation functions that used in the neural network to activate the status of the neuron are ZMAF, sigmoid function or HTAF. In this study, there are two types of activation functions that are studied and evaluated by computer simulation. HTAF and ZMAF will be integrated into HNN and the effectiveness between these two activation functions will be analyzed by the computer simulations. The main objective for analyzing the effectiveness of these two activation functions is to decide which activation function will produce optimal solutions in the HNN for doing logic program. Different types of activation functions showed in the below Figure 2 and there are advantages and disadvantages of various activation functions are given in table 1.

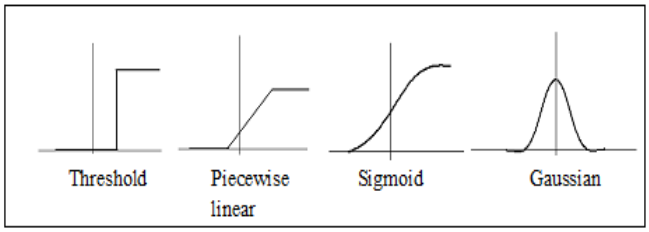

Figure 2: Differents types of Activation Function [3]

Table 1. List of advantages and disadvantages of activation functions [23, 24]

Advantages	Disadvantages
HTAF is easier to understand and is used mostly in networks. HTAF produces zero centered output thereby aiding the back-propagation process.	HTAF vanishing gradient problem.

ZMAF doesn't stimulate all the neurons at the same time. Because the output of some neurons is zero so therefore only some neurons are stimulated in the network, which makes the network sparse, efficient, and easy for computation.	ZMAF needs to go through the adjustable parameters to manage the threshold for the sensitive area.

II.3.1 Hyperbolic Tangent Activation Function (HTAF)

HTAF is used in many ANNs due to the role of this function that can prevent the network from collapsing in simplistic linear functions. This enable generated activation function defined as the output of the status neuron as (-1 or 1) [14]. Update the state of the neuron in-network by using the following equation:

$$S_i(t+1) = \operatorname{sgn}\left[g\left(h_i(t)\right)\right] \tag{5}$$

whereas $h_i(t)$ is called local field of HNN. The equation of HTAF as followed [25]:

$$g(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}} \tag{6}$$

The neuron status will be updated as follows:

$$S_i(t+1) = \begin{cases} 1 \;\; for \;\; g(h_i) \geq 0 \\ -1 \;\; for \;\; g(h_i) < 0 \end{cases} \tag{7}$$

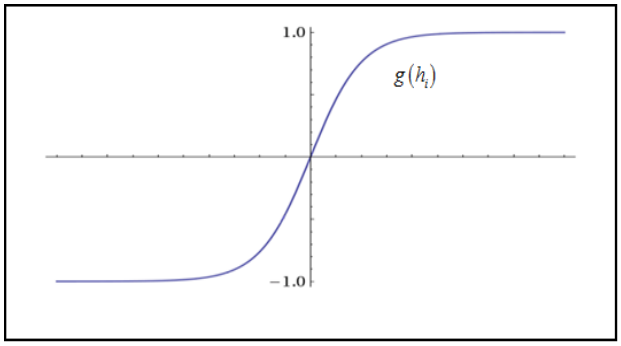

Figure 3. Hyperbolic Tangent Activation Function (HTAF) [25]

II.3.2 Zeng Martinez Activation Function (ZMAF)

ZMAF in HNN is defined as the sigmoid function as shown in the equation (8). Nonetheless, this activation function focuses on insignificant noise disturbances rather than on cost-related signals and network-encoded constraints.

$$V_{x_i} = \begin{cases} \dfrac{0.5\left(1+\tanh\left(\dfrac{U_{x_i}+x_0}{u_0}\right)\right)}{1+\tanh\left(\dfrac{x_0}{u_0}\right)} & ,\left(U_{x_i} < 0\right) \\ \dfrac{\tanh\left(\dfrac{x_0}{u_0}\right)+0.5\left(1+\tanh\left(\dfrac{U_{x_i}-x_0}{u_0}\right)\right)}{1+\tanh\left(\dfrac{x_0}{u_0}\right)} & ,\left(U_{x_i} \geq 0\right) \end{cases} \tag{8}$$

where U_{x_i} is initial states, V_{x_i} is activation function, x_0 symbolizes the threshold for V_{x_i}, and u_0 is measurement the steepness of ZMAF. Basically, the input of the neurons is updated according to the following equation:

$$S_i(t+1) = \operatorname{sgn}\left[g\left(h_i(t)\right)\right] \tag{9}$$

whereas $h_i(t)$ is local field of HNN. ZMAF is in the style of $g(h)$ as followed:

$$g(h_i) = \begin{cases} \dfrac{0.5\left(1+\tanh\left(\dfrac{h_i+x_0}{u_0}\right)\right)}{1+\tanh\left(\dfrac{x_0}{u_0}\right)} & ,\left(h_i < 0\right) \\ \dfrac{\tanh\left(\dfrac{x_0}{u_0}\right)+0.5\left(1+\tanh\left(\dfrac{h_i-x_0}{u_0}\right)\right)}{1+\tanh\left(\dfrac{x_0}{u_0}\right)} & ,\left(h_i \geq 0\right) \end{cases} \tag{10}$$

The neuron state updating rules are integrated with ZMAF which is given in equation (11). This feature will withstand noise when the network becoming larger [8, 9].

$$S_i(t+1) = \begin{cases} 1 \;\; for \;\; g(h_i) \geq 0 \\ -1 \;\; for \;\; g(h_i) < 0 \end{cases} \tag{11}$$

II.4 Implementation of Activation Functions for Doing Logic Programming in HNN

In general, global solution of HNN can be affected by how the network acts as a "mapping technique". In this case, the "mapping technique" is the activation function. Incorporated with the right activation function, the well built network will generate efficient and effective performance. According to Zeng & Martinez in their different papers [8, 23], one of the most significant factors affecting the quality of solution is disparity in updating rules when input is provided by a specific neuron. This study implemented two new activation functions, HTAF and ZMAF, in order to achieve the optimum results. Below are the steps involved in implementing the activation function in HNN logic programming:

1. Translate all HornSAT logic programs into simple Boolean algebraic form Conjunctive Normal Form (CNF) with logic program given in the following equation:

$$P = \left(E \vee \neg F \vee \neg G\right) \wedge \left(B \vee \neg C\right) \wedge \left(A\right) \tag{12}$$

2. Specify each neuron to all its ground.

3. Initiate each of the synaptic weights of the neurons to 0.

4. Determine the cost function where the neuron represented by the following equation:

$$V_x = \frac{1}{2}(1+S_x) \tag{13}$$

and the negation of the neuron given by the following equation:

$$\overline{V_x} = \frac{1}{2}(1-S_x) \tag{15}$$

where V_x symbolizes the logic value of the neuron *x*, whereas S_x is the status of the neuron *x* which is given by the following equation:

$$S_x \in \{1,-1\} \tag{14}$$

The logical connective form of a conjuncture in equation (12) is designed by a multiplication, while the negation of the equation (12)

will be in the connective form of a disjunction (DCF) as show in the following equation (16) :

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$\neg P = (\neg E \wedge F \wedge G) \vee (\neg B \wedge C) \vee (\neg A) \quad (16)$$

5. Calculate the synaptic weight by comparing b the cost function C_p in equation (17) and the energy function E_p in equation (4) by utilizing Wan Abdullah's method.

$$C_p = \left[\begin{array}{l} \frac{1}{8}(1-S_E)(1+S_F)(1-S_G)]+ \\ \frac{1}{4}(1-S_B)(1+S_C)]+\frac{1}{2}(1-S_A) \end{array} \right] \quad (17)$$

6. Applied Sathasivam's relaxation technique by the following equation (18) to assure HNN is relaxed to help the neurons to reach the final state [19]:

$$\frac{dh_i^{new}}{dt} = R\frac{dh_i}{dt} \quad (18)$$

whereas h_i refers to the local field of HNN and R indicates the rate of relaxation.

7. Randomise the state of neuron to provide chance to network to be satisfiabile with learned interpretation and this interpretation will be stored.

8. Use the equation (2) to find the corresponding local field.

9. Using HTAF or ZMAF to identify and update the final state of the neurons.

10. Compute the final energy for the stable state.

11. If the state of the neuron in HNN unchanged for more than 5 runs, it is classified as a stable state.

12. Compute the final state of the output of the neurons in HNN.

In the next section, Agent-Based Modeling (ABM) will be developed by utilizing NETLOGO as the platform to carry out logic programming (LP) in Hopfield Neural Network (HNN) with different activation functions such as ZMAF and HTAF.

II.5 AGENT BASED MODELLING

NETLOGO software was developed and edited by "Uri Wilensky" who is the director of the "Center for Connected Learning and Computer-Based Modelling" as Agent-based modeling at Northwestern University [11]. Agent-based modelling (ABM) is a methodology widely utilized in a wide area of social sciences [10]. This requires the development of a computer model consisting of "agents" representing both a social entity and a "world," NETLOGO being an agent-based language for programming and an essential framework for modeling. NETLOGO was planned to be a "low threshold and no ceiling" in the spirit of logical programming. NETLOGO teaches programming concepts in tortoises, patches, connections and observers using agents [12]. NETLOGO was designed for a variety of publics, particularly: educational children and field experts without programming history to model-related phenomena. NETLOGO has been used in many academic publications [12]. When connecting agents to mobile devices, they build networks, charts and aggregates to allow users to gain a greater understanding of system performance. In fact, the runs are exactly cross-platform reproducible. An agent-based simulation is used to model the interaction of HNN to do LP by utilized different activation functions such as the ZMAF and HTAF. Agent-Based Modeling uses NETLOGO as a platform. A flow chart in Figure 4 shows how ABM developed for comparing the activation functions.

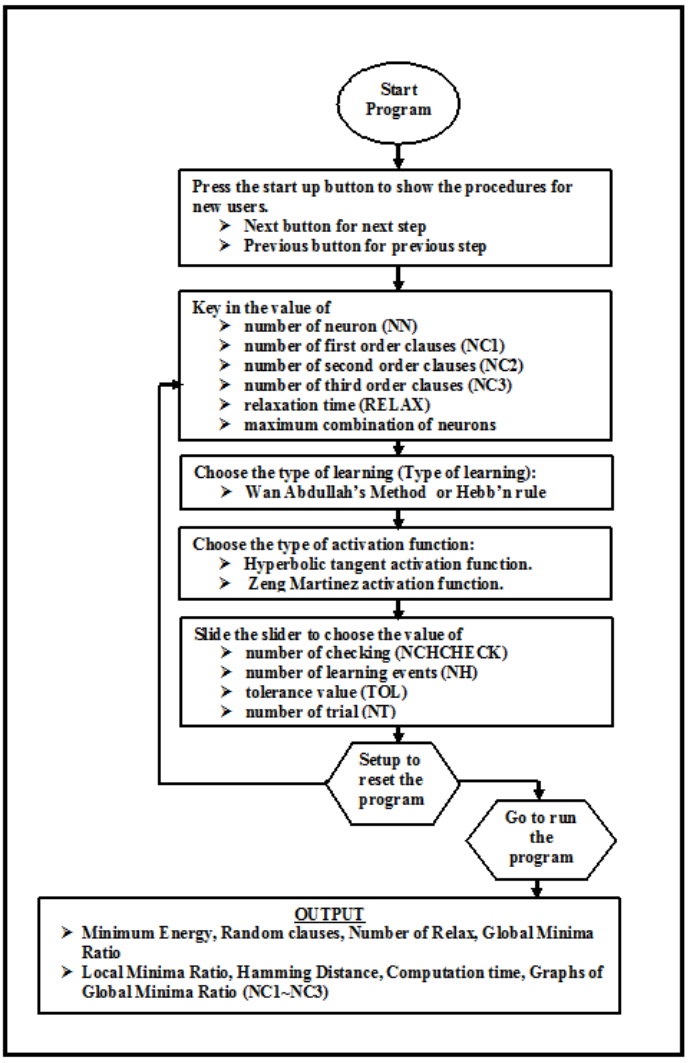

Figure 4: Flowchart of Agent-Based Modeling of HNN with HTAF or ZMAF [7]

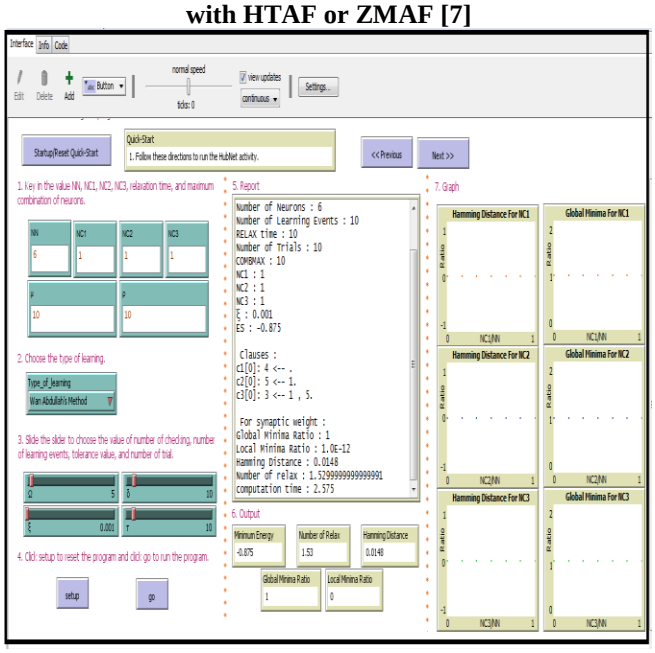

Figure 5: The screenshot of the NETLOGO software for doing Agent-Based Modeling

Figure 5 is explained as follows:

STAGE 1: Entering the Value of Each Parameters

1. Press the startup / Reset Quick-Start button for the new user. Users will click the next button to take the next step and the last step.

2. In this step enter in the button the value of *NN*, NC1, NC2, NC3, and setups the value of Relax μ, value of COMBMAX ρ.

3. In this step choose types of activation function and learning.

4. Next slides the slider to settle number of trial τ, number of learning event δ . The maximum tolerance value, ξ is set to 0.001 by try and error technique.

5. After inserting the values, press the setup button in the program to set these values.

6. Finally, press the button "go" to run the program which will start by generating randomly the programming clauses.

STAGE 2: Training

7. Initializing initial states of neuron in each clauses. Based on the design of HNN that derived from the behaviors of neurons movement in the magnetic domain, all neuron will communicate with each other in a complete bonded form as all of them tries to reach active appropriate state and this means minimal energy function. This stage is called as activation by using the activation functions. In this state the neurons will be rotating. Thus, let the network evolve to the minima capacity when the minima energy equals the energy prerequisite to achieve the global minima.

8. The neurons relaxed to the stable states if the final states remained unchanged after five runs.

9. Next, compute the final energy based on the stable state.

10. Accept the final solution as a global solution if the difference between the minima energy and the final energy is within tolerance, or else go back to step1.

11. Finally, compute the ratio of global solutions and print out the results as shown in Figure 5.

III. RESULTS AND DISCUSSION

III.1 Experimental

Models have been developed using software NETLOGO [10] version 6.1.1, which is a feature of a state-of - the-art device and buttons that minimize the length of the program. The global minimal ratio, hamming distance, and calculation time were calculated by using the computer simulations for the both activation functions Zeng Martinez Activation Function and Hyperbolic Tangent Activation Function. We examined all levels of the provisions including NC1, NC2 and NC3 ($1 \le NC_i \le 15$) for different combination of neurons ($6 \le NN \le 90$). For each of these combinations, all value of the parameters are chosen by try and error technique by using different setup of number of learning events δ, the value of number of checking Ω, tolerance value ξ, Relax μ, COMBMAX ρ, number of trial τ. The outputs of the simulations are Global Minima Ratio (*zM*), Hamming Distance (*HD*), and *CPU time*. These will be helping us in endorsing the performance of both of the activation functions HTAF and ZMAF for doing logic program in HNN. Subsequently, the parameters for the both activation functions are summarized systematically as in Table 1 and Table 2.

Table 1. List of Parameters

Parameter	Parameter Value
Ω	5
δ	10
ξ	0.001
τ	10
μ	10
ρ	10

Table 2. List of Parameters

Parameter	Parameter Value
Ω	5
δ	100
ξ	0.001
τ	100
μ	100
ρ	100

III.2 Discussion

A. Global minima ratios (*zM*):

zM is the ratio between global total minima energy and total simultaneous numbers is defined by Kasihmuddin *et al* and Alzaeemi *et a l*[19, 20]. When the final energy is within the limit, it is known as global minimum energy. The global minima ratio equation is defined as:

$$zM = \frac{1}{tc}\sum_{i=1}^{n} N_p \tag{19}$$

The local minimum ratio equation is defined as [21]:

$$L_m = 1 - zM \tag{20}$$

where *c* is the combination of the neuron, *t* is the trial, and N_p is the number of global minima energy of the propose model. The higher accuracy HNN model has higher value of zM .

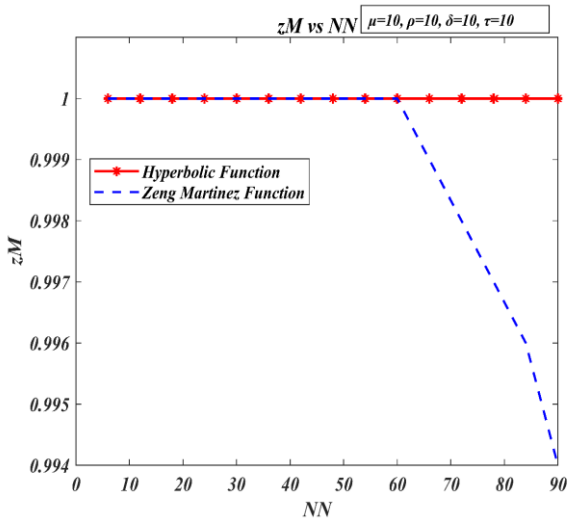

Figure 6: Global minimum ratio for HTAF and ZMAF for μ=10, ρ=10, δ=10, τ=10

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

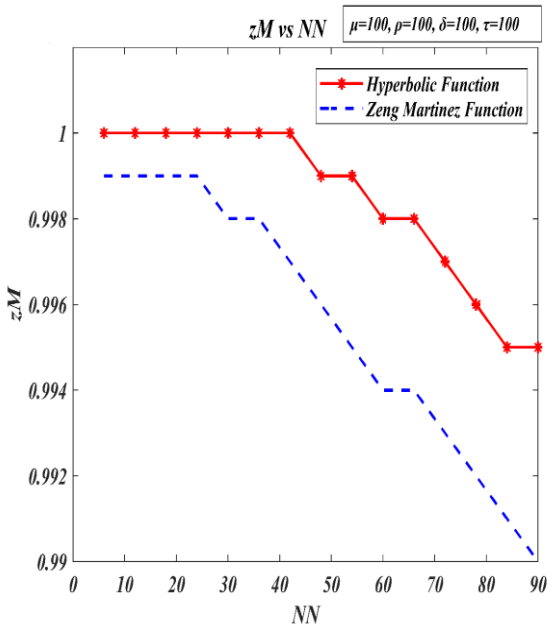

Figure 7: Global minimum ratio for HTAF and ZMAF for μ=100, ρ=100, δ=100, τ=100

In this paper we implemented the ZMAF and HTAF to stimulate the performance of doing logical program in HNN. Figure 6 and Figure 7 shows the results for *zM* obtained for both activation functions for doing LP in HNN based on Wan Abdullah's method for the different number of literals per clauses, different number of neurons and different values of setups of number of learning events δ, number of trial τ, Relax μ, and COMBMAX ρ. We can observe that the ratio of global solutions *zM* for HTAF is equal to 1 when using the smallest value of the parameters, δ=τ=μ=ρ=10 as shown the Figure 6 or closer to 1 when using the highest value of these parameters, δ=τ=μ=ρ=100 as shown in the Figure 7. We can observed that HTAF outperformed ZMF even when the network gets larger and complex. This indicates that the output by HTAF in HNN is almost all solutions are "correct". It can be observed that when the network gets larger or complex, the hyperbolic tangent activation function still can accommodate more neurons and produce a good output. Almost all the solutions obtained by using the HTAF are global minimum solutions.

B. Hamming Distance (*HD*):

Hamming Distance has defined as the dimension between the global solution and the stable state of the neurons [26].

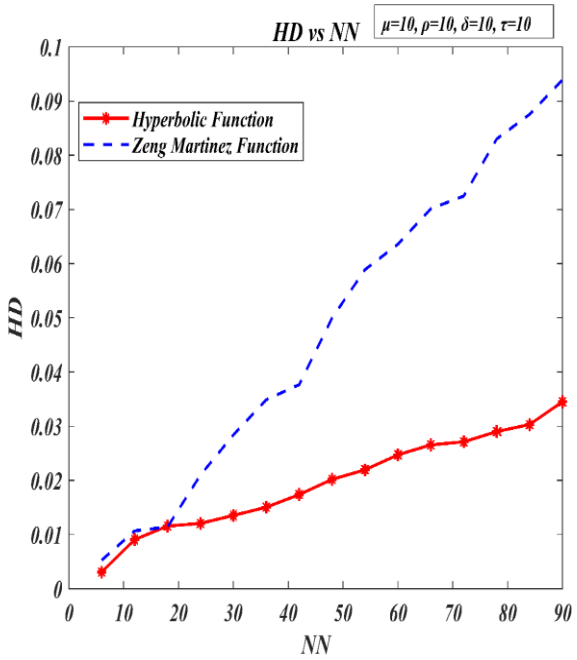

Figure 8: *HD* for HTAF and ZMAF for μ=10, ρ=10, δ=10, τ=10

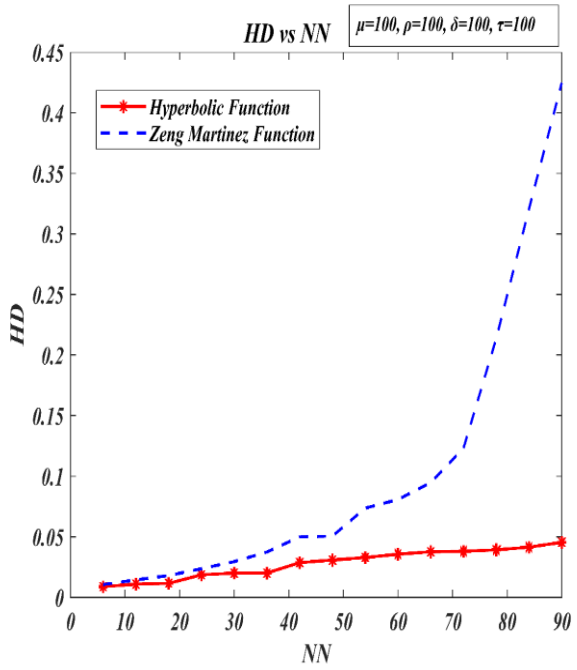

Figure 9: *HD* for HTAF and ZMAF for μ=100, ρ=100, δ=100, τ=100

Figure 8 and Figure 9 show the *HD* for doing LP in HNN by using different activation functions as HTAF and ZMAF. It can be observed that the HD for HTAF is equal to 0 or closer to 0 compared to ZMAF despite of increased in the complexity of the network. As the network complexity increased, HTAF outperformed ZMAF in terms of HD. Hence, the training process in HNN by using HTAF accelerated better than ZMAF.

C. *CPU time*:

In this study, we define the *CPU time* as the total time that is taken for the network to generate maxima satisfied clauses in the logic programming by using

Retrieval Number: D7711049420/2020©BEIESP
DOI: 10.35940/ijeat.D7711.049420
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

different activation functions [13, 14]. *CPU time* is given by the following equation [22]:

$$CPU\ time = Retrieval\ time + Learning\ time \quad (21)$$

As defined by Sathasivam & Abdullah [22], the best model HNN has the least *CPU time* during retrieval phase and the learning phase. Hence, the best HNN model with the different activation functions will have the shortest *CPU time*.

From Figure 10 and Figure 11 shows the *CPU time* acquired in doing LP in HNN by using different activation functions as HTAF and ZMAF. Figure 10 and Figure 11 shows that the *CPU time* of HTAF outperforms the *CPU time* of ZMAF. HTAF is preferred compared to ZMAF especially when the network gets complex because by using HTAF the network manages to find the global minima quickly due to the derivatives of HTAF are shorter than ZMAF. In other words, the cost function is minimized faster when HTAF is utilized.

From the results above, we can conclude that HTAF is better than ZMAF for doing LP in HNN. The estimated output or restricted properties of HTAF supports the network in providing good outputs. Due to this, the output the model reaches solutions faster.

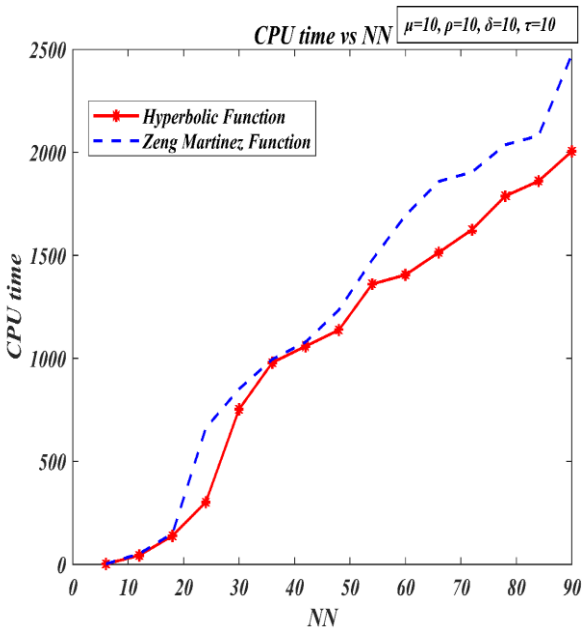

Figure 10: *CPU time* for HTAF and ZMAF for μ=10, ρ=10, δ=10, τ=10

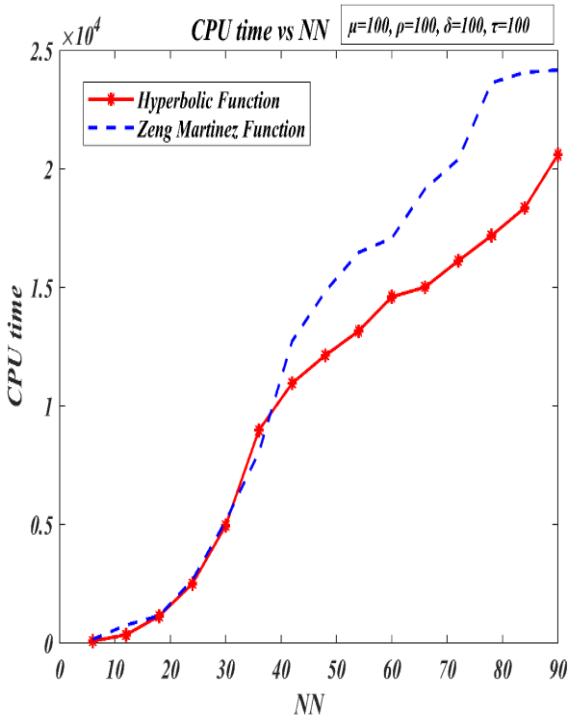

Figure 11: *CPU time* for HTAF and ZMAF for μ=100, ρ=100, δ=100, τ=100

IV. CONCLUSION

In this paper, we had developed agent-based modeling (ABM) to model the acceleration ability of the two activation functions: Hyperbolic Tangent Activation Function (HTAF) and Zeng Martinez Activation Function (ZMAF) for doing logic programming (LP) in Hopfield Neural Network (HNN) by using NETLOGO as the platform. ABM presents a natural explanation of the system to be ready to integrate/link of activation function to do LP in HNN. From the experimental results obtained for global minima ratio, hamming distances and computation time, HTAF outperformed ZMAF in doing LP in HNN. Various selections of setups parameters for the number of learning events δ, the number of trial τ, Relax μ, COMBMAX ρ were used for the simulation. The results obtained verified our proposed theory that HTAF outperformed ZMAF.

ACKNOWLEDGMENT

This research is supported by Research University Grant (RUI)(1001/PMATHS/8011131) by Universiti Sains Malaysia.

REFERENCES

1. W. A. T. W. Abdullah, "Logic programming on a neural network." *International journal of intelligent systems* 7.6, 1992, pp. 513-519.
2. S. Sathasivam, and M. Velavan, "Agent Based Modelling For New Technique In Neuro Symbolic Integration." *Matter: International Journal of Science and Technology* 3.2, 2017, pp. 1-10.
3. K. Bekir and A. O. Vehbi, "Performance analysis of various activation functions in generalized MLP architectures of neural networks." *International Journal of Artificial Intelligence and Expert Systems* 1.4, 2011, pp. 111-122.

4. S. A. Alzaeemi. *et al.*, "Analysis of Performance of Various Activation Functions for doing the logic programming in Hopfield Network.". *Journal of Computational Bioinformatics and in Silico Modeling*, 6, 2, 2017, pp. 911-921.
5. G. Garg, and S. Poonam. "An Analysis of Contrast Enhancement using Activation Functions'." *International Journal of Hybrid Information Technology* 7.5, 2014, pp. 234-244.
6. P. M. M. R. Pushpa, and K. Manimala, "Implementation of hyperbolic tangent activation function in VLSI." *International Journal of Advanced Research in Computer Science & Technology* 2, 2014, pp. 225-228.
7. S. Sathasivam, and N. Pei Fen, "Developing agent based modeling for doing logic programming in hopfield network." *Applied Mathematical Sciences* 7.1, 2013, pp. 23-35.
8. X. Zeng, and T. R. Martinez, "A new activation function in the Hopfield Network for Solving Optimization Problems." *Artificial Neural Nets and Genetic Algorithms*. Springer, Vienna, 1999, pp. 73-77.
9. S. Sathasivam, "Usage of New Activation Function in Neuro-Symbolic Integration." *AIP Conference Proceedings*. Vol. 1325. No. 1. American Institute of Physics, 2010, pp. 171-174.
10. M. Lippe, *et al.* "Using agent-based modelling to simulate social-ecological systems across scales." *GeoInformatica* 23.2, 2019, pp. 269-298.
11. Y. Dijkxhoorn, *et al*. "Trusted sorghum: simulating interactions in the sorghum value chain in Kenya using games and agent-based modelling." *Journal of Development Effectiveness* 11.2, 2019, pp. 146-164.
12. C. Delcea, *et al*. "An agent-based modeling approach to collaborative classrooms evacuation process." *Safety science* 121, 2020, pp. 414-429.
13. W. A. T. W. Abdullah, "The logic of neural networks." *Physics Letters A* 176.3-4, 1993, pp. 202-206.
14. S. Sathasivam, "Upgrading logic programming in Hopfield network." *Sains Malaysiana* 39.1, 2010, pp.115-118.
15. S. Sathasivam, *et al*, "Hybrid Discrete Hopfield Neural Network based Modified Clonal Selection Algorithm for VLSI Circuit Verification." *Pertanika Journal of Science & Technology* 28.1,2020.
16. L.C. Kho, *et al.* "Logic Mining in League of Legends." *Pertanika Journal of Science & Technology* 28.1, 2020.
17. S. A. Alzaeemi, *et al.* "Comparing the logic programming between Hopfield neural network and radial basis function neural network." *AIP Conference Proceedings*. Vol. 2184. No. 1. AIP Publishing LLC, 2019.
18. M. S. M. Kasihmuddin, *et al*. "Discrete Mutation Hopfield Neural Network in Propositional Satisfiability." *Mathematics* 7.11, 2019, pp.1133-1154.
19. M. S. M. Kasihmuddin, *et al.* "Quality Solution of Logic Programming in Hopfield Neural Network." *Journal of Physics: Conference Series*. Vol. 1366. No. 1. IOP Publishing, 2019.
20. A. Alzaeemi, and S. Sathasivam, "Hopfield neural network in agent based modeling". *MOJ App Bio Biomech*, 2(6), 2018, pp. 334-341.
21. M. S. M. Kasihmuddin, *et al*. "Discrete Hopfield Neural Network in Restricted Maximum k-Satisfiability Logic Programming." *Sains Malaysiana* 47.6, 2018, pp. 1327-1335.
22. S. Sathasivam, & W. A. T. W. Abdullah, "Logic mining in neural network: reverse analysis method." *Computing* 91.2, 2011, pp. 119-133.
    X. Zeng, and T. R. Martinez, "A New Relaxation Procedure in the Hopfield Network for Solving Optimization Problems." *Neural Processing Letters* 10.3, 1999, pp. 211-222.
23. V. Kathirvel, *et al*. "Hybrid imperialistic competitive algorithm incorporated with hopfield neural network for robust 3 satisfiability logic programming." *IAES International Journal of Artificial Intelligence* 8.2, 2019, pp. 144.
24. M. S. M. Kasihmuddin, *et al*. "Students' performance via satisfiability reverse analysis method with Hopfield Neural Network." *AIP Conference Proceedings*. Vol. 2184. No. 1. AIP Publishing LLC, 2019.
25. M. Islam, *et al.* "Performance comparison of various probability gate assisted binary lightning search algorithm." *IAES International Journal of Artificial Intelligence*, 2019, pp. 228-236.

AUTHORS PROFILE

Shehab Abdulhabib Alzaeemi received a Bachelor Degree of Education (Science) from Taiz Universiti in 2004 and Master of Science (Mathematics) from Universiti Sains Malaysia in 2016 and now student PhD in Universiti Sains Malaysia. He was fellow under the Academic Staff Training System of Sana'a Community College from 2005-2014. His research interests mainly focus on neural network, logic programming, and data mining.

Saratha Sathasivam is an Associate Professor in the School of Mathematical Sciences, Universiti Sains Malaysia. She received her MSc and BSc(Ed) from Universiti Sains Malaysia. She received her Ph.D at Universiti Malaya, Malaysia. Her current research interest are neural networks, agent based modeling and constrained optimization problem.

Muraly Velavan is a senior lecturer and also the Deputy Director in the School of General & Foundation Studies, AIMST University. He received his MSc at Universiti Malaysia Perlis. His current research interest are neural networks and agent based modeling.

Mustafa bin Mamat is a Professor in the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin. His current research interests include computational mathematics, convergence analysis and evolutionary algorithm.