International Journal of Computer Networks and Communications Security

VOL. 7, NO. 12, DECEMBER 2019, 250–257
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

Agent Based Modeling for Comparing the Performances of Hyperbolic and Zeng and Martinez Activations Functions

Mohammad Athar Azim[1], Saratha Sathasivam[2], Shehab Abdulhabib Saeed Alzaeemi[3] and Maqsood Mahmood[4]

[1, 2, 3, 4] School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang Malaysia

[1]*abuaayan1973@gmail.com*

ABSTRACT

Agent based modeling is widely emerging phenomenon used for comparing the performances of activation functions. There are many customary activation functions which we are using with neural networks (NNs). The appearance of activation functions for artificial neural networks (ANNs) is used as system functions in research and technology. The most common way for the use of this demand were its boundedness in the unit interval, the functions, and its derivative's fast computability, and a number of amenable mathematical properties in the realm of approximation theory. The sole reason of this research paper is to develop the most effective activation function in doing logic programming in the background of Hopfield network. A correlation is carried out between hyperbolic tangent activation functions and (Zeng & Martinez) function based on Wan Abdullah method. These evaluations are done on the basis of global minima ratio, hamming distance and computational time. Moreover computer simulations using software NETLOGO 5.3.1 are carried out to compare the effectiveness of these two activations functions.

Keywords: *Logic Programming Hopfield network, Wan Abdullah's method, Zeng and Martinez function, Hyperbolic Tangent activations function.*

1 INTRODUCTION

Neaural network is a mathematical model or a parallel processing network, whom is inspired by the biological structure of neaurans as information of brain process.To express the associated knowledge of input and output,It uses the idea of non linear mapping and the method of parallel processing. We will design an agent based modeling to implement the Hopfield network in doing logic programming. The first transfer function that can be eliminated in logic programming within Hopfield neural network is the sign function of McCulloch-Pitts (ideal model) proposed by Wan Abdullah [1, 2]. The activation function is defined as the output of the neuron by the given input [4]. The most common for use of activation functions are logistic sigmoid function and Gaussian basis functions[5]. The main argument of this paper is the study of hyperbolic tangent activation function and Zeng function. Detail evaluation (computer simulation) is carried out to analyze the validness of these functions. Therefore, the purpose of this dissertation is to compare the performance of the following activation functions: Hyperbolic tangent activation function and Zeng activation function.

This paper is organized as follows, in section 2, an outline of Hopfield network is given and method of doing logic programming in neural network is described. In section 3, Types of activation functions [Hyperbolic and Zeng and Martinez] are described. In section 4, Regarding Net Logo and agent based modeling. In section 5 ,about the Simulations and discussion about the Global minima[ZM], Hamming distance[HD] and Computation time[CT]. And finally in section 6 and 7 occupy the discussion and concluding remarks regarding this work.

2 HOPFIELD NEURAL NETWORK AND LOGIC PROGRAMMING HOPFIELD NEURAL NETWORK

Hopfield network uses the formal neuron of McCulloch-Pitts, so the neuron in Hopfield network has two versions either binary or continuously valued. For example, the units only take on two different values for their states and the value is determined by whether or not the units' input exceeds their threshold. Hopfield network can either have units that take on values of 1 or -1, $S_i \in \{-1, 1\}$, but for continuous networks, S_i can be any value between 0 and 1. For a network consist of N neurons, the definition of the state of the network is as follows:

$$S = [S_1, S_2, \dots, S_N]^T$$

The induced local field, h_i of neuron *i* is defined by

$$h_i = \sum_j J_{ij} S_j + J_i$$

Where

- J_{ij} is the strength of the synaptic strength from neuron j to neuron i
- S_j is the state of neuron j.
- $-J_i$ is the threshold of neuron *i*.

Hence, the neurons obey the dynamics

$$S_i \rightarrow sgn(h_i)$$
(2.3)

Where S_i *is the state of neuron i while, sgn* is the signum function.

The connections in a Hopfield network typically have the following restrictions:

- $J_{ii} = 0$, $\forall i$ (no self-feed-back connections) (2.4)
- $J_{ij} = J_{ji}$, $\forall i,j$ (connections are symmetric) (2.5)

The following algorithm summates how a logic program can be carried out in a Hopfield network based on proposal by Wan Abdullah [1]whom is recognize as Direct method [1, 2]:

(i) Given a logic program, translate all the clauses in the logic program into basic Boolean algebraic form.
(ii) Identify a neuron to each ground neuron.
(iii) Initialize all connections strength to zero.
(iv) Derive a cost function that is associated with the negation of all the clauses, such that *½(1+S_x)* represents the logical value of a neuron *X*, where S_x is the neuron corresponding to *X*. The value of S_x is defined in such a way that it carries the values 1 if *X* is true and -1 when *X* is false. Negation (neuron *X* does not occur) is represented by *½(1-S_x);* a conjunction logical connective is represented by multiplication whereas a disjunction connective is represented by addition.
(v) Obtain the values of connection strength by comparing the cost function with the energy, *E*. *E* is defined as below where: J_i, J_{ij}, J_{ijk} are the synaptic strength of neuron *i*, from neuron *j* to neuron *i* , from neuron *k* to *i* respectively. While S_i , S_j , S_k are the state of neuron *i, j,* and *k* respectively.

$$E = -\frac{1}{3}\sum_i \sum_j \sum_k J^{(3)}_{[ijk]} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j J^{(2)}_{[ij]} S_i S_j - \sum_i J^{(1)}_{[i]} S_i \qquad (1)$$
(2.1)

(vi) Let the neural networks evolve until minimum energy is reached. Check whether the solution obtained is a global solution which is difference between global minimum energy and final energy is within the tolerance value [9]. (2.2)

3 TYPES ACTIVATION FUNCTION

Bekir and Vehbi [3] suggested that the most important unit in neural networks structure is their net inputs using a scalar-to-scalar function, which called "the activation function or threshold function or transfer function", and their net outputs (called the unit's activation) a result value. An activation function is limiting the amplitude of the output of a neuron. They [3, 4] also advised that the action of an artificial neural network is to sum the product associated with weight and the input signal and produce an output or activation function. The most accepted for the use of activation functions are: Zeng function and hyperbolic tangent activation function.

3.1 Hyperbolic Tangent Function

The hyperbolic tangent activation function is the most accepted activation function for neural network. It range is between -1 and 1 and most useful for training data that is also between values of 0 and 1. The hyperbolic tangent activation function has a derivative that can be used with gradient descent based training methods [6].

The function is given by:

$$g(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \qquad (1)$$

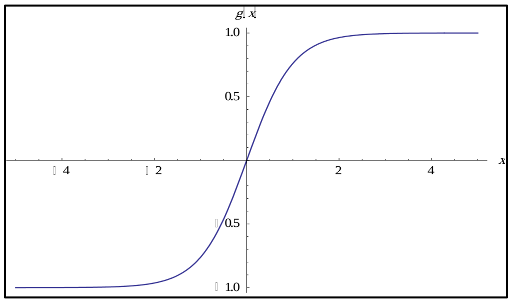


Fig. 1:. Hyperbolic Tangent function [6]

3.2 Zeng and Martinez Activation Function

The activation function in the Hopfield network is the sigmoid function (equation 4). However this activation function puts too much emphasis on minor noise perturbation instead of the signals related to the cost and the constraints encoded in the network. Zeng and Martinez [8] proposed a new activation function as followed:

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

(2.7)

where the parameters are defined as followed: V_{x_i} = activation function, U_{x_i} =initial states, = represents the threshold for V_{x_i} to become steep, and u_0 measures the steepness of the activation function. This function can tolerate with noise and do perform well when the network gets larger [8, 9].

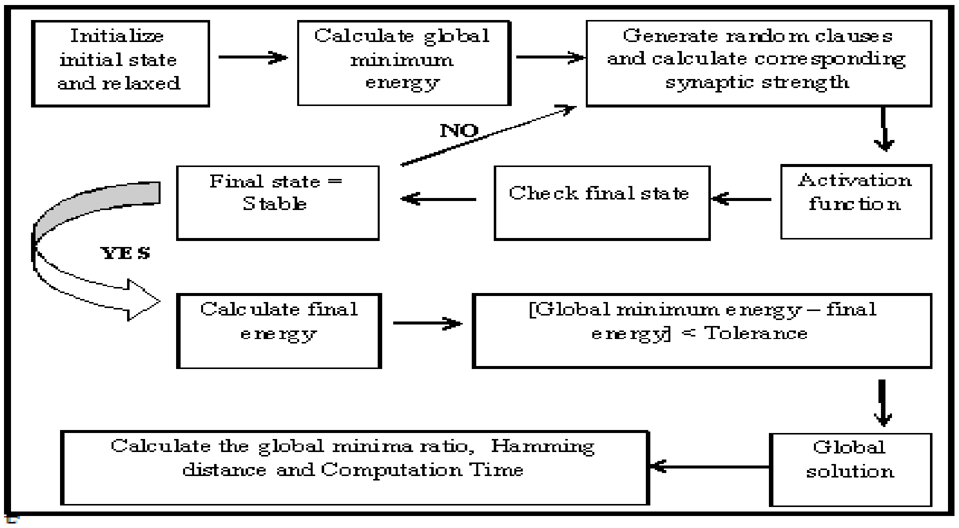

Initialize initial state and relaxed
Calculate global minimum energy
Generate random clauses and calculate corresponding synaptic strength
NO
Final state = Stable
Check final state
Activation function
YES
Calculate final energy
[Global minimum energy – final energy] < Tolerance
Calculate the global minima ratio, Hamming distance and Computation Time
Global solution

Fig. 2. Flow Chart on the Implementation of Hyperbolic Tangent Activation Function and Zeng and Martinez Activation Function

4 AGENT BASED MODELLING

After knowing the efficiency of doing logic programming in Hopfield network, NETLOGO is used as a platform to develop agent based modelling (ABM) [7]. We will design an agent based modelling to implement the Hopfield network in doing logic programming. Net logo was designed and authored by Uri Wilensky, director of Northwestern University's Center for Connected Learning and Computer-Based Modelling. Agent-based modeling (ABM) is a technique increasingly used in a broad range of social

sciences [10]. It involves building a computational model consisting of "agents," each of which represents an actor in the social world, and an" environment"[34]Netlogo is an agent based programming language and integrated modeling environment. Net logo was designed, in the spirit of the Logo programming language, to be "low threshold and no ceiling". It teaches programming concepts using agents in the form of *turtles*, *patches*, *links* and the *observer*. Netlogo was designed for multiple audiences in mind, in particular: teaching children in the education community and domain experts without a programming background to model related phenomena. Many scientific articles have been published using Netlogo. While for the link agents, they connect the mobile agents to make networks, graphs and aggregates which can let the users get more understanding on output of system. Moreover, its runs are exactly reproducible cross-platform. In the next chart will carry out the definition and more explanation of simulator and benefits of agent based modelling in NETLOGO.

Netlogo will be used as a platform to develop agent based modelling (ABM). Agent based modelling designed to implement Hopfield network in doing logic programming. Netlogo is a multi-agent programming language and integrated modelling environment as shown in Figure 3. Figure 2: Flow Chart of Agent Based Modelling.

5 EXPERIMENTAL / SIMULATIONS AND DISCUSSION

The computer simulations were proved by using NETLOGO [10] software version 5.3.1 which was symbolized modern tool and buttons which decreases the duration of the program was used fundamental in gaining the results. The global minima ratio, Hamming distance and computation time for the hyperbolic tangent activation function and Zeng function were attained experimentally from the computer simulations. For each of the data, we considered every level of clauses such as *NC1, NC2* and *NC3* (5, 10, 15, 20, 25 and 30) for different number of neurons (10, 20, 30, 40, 50, 60, 70 and 80). These outputs helped us in confirming the performance of each of the hyperbolic tangent activation function and Zeng function in doing logic program in Hopfield network.

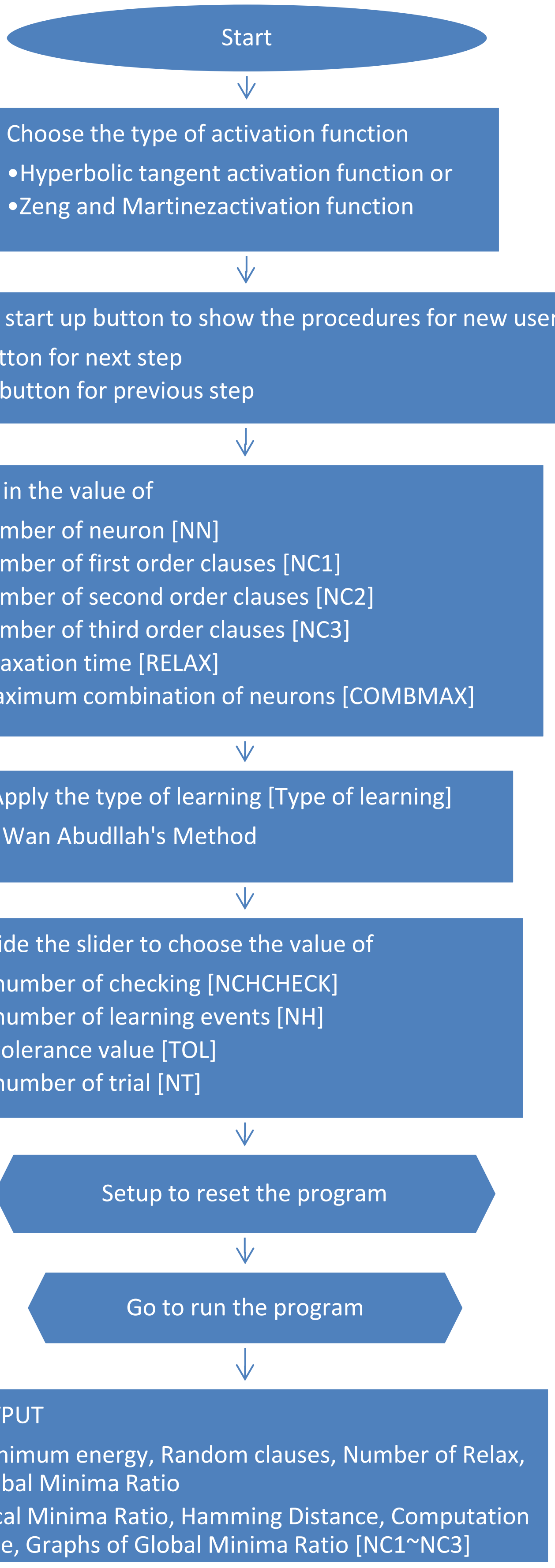
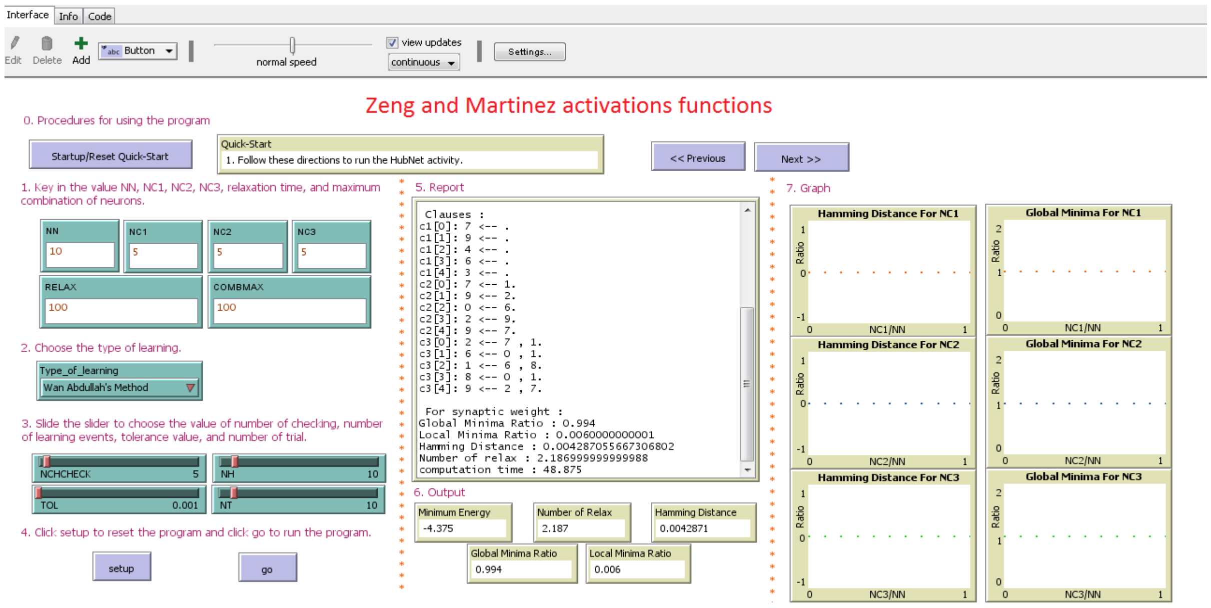

Start
Choose the type of activation function
•Hyperbolic tangent activation function or
•Zeng and Martinezactivation function
Press the start up button to show the procedures for new users.
•Next button for next step
•Prvious button for previous step
Key in the value of
•number of neuron [NN]
•number of first order clauses [NC1]
•number of second order clauses [NC2]
•number of third order clauses [NC3]
•relaxation time [RELAX]
•maximum combination of neurons [COMBMAX]
Apply the type of learning [Type of learning]
•Wan Abudllah's Method
Slide the slider to choose the value of
•number of checking [NCHCHECK]
•number of learning events [NH]
•tolerance value [TOL]
•number of trial [NT]
Setup to reset the program
Go to run the program
OUTPUT
•Minimum energy, Random clauses, Number of Relax, Global Minima Ratio
•Local Minima Ratio, Hamming Distance, Computation time, Graphs of Global Minima Ratio [NC1~NC3]

Fig. 2. Flow Chart of Agent Based Modelling



0. Procedures for using the program

Quick-Start
1. Follow these directions to run the HubNet activity.

1. Key in the value NN, NC1, NC2, NC3, relaxation time, and maximum combination of neurons.

NN	NC1	NC2	NC3
10	5	5	5

RELAX	COMBMAX
100	100

2. Choose the type of learning.

Type_of_learning: Wan Abdullah's Method

3. Slide the slider to choose the value of number of checking, number of learning events, tolerance value, and number of trial.

NCHCHECK 5	NH 10
TOL 0.001	NT 10

4. Click setup to reset the program and click go to run the program.

5. Report

```
 Clauses :
c1[0]: 7 <-- .
c1[1]: 9 <-- .
c1[2]: 4 <-- .
c1[3]: 6 <-- .
c1[4]: 3 <-- .
c2[0]: 7 <-- 1.
c2[1]: 9 <-- 2.
c2[2]: 0 <-- 6.
c2[3]: 2 <-- 9.
c2[4]: 9 <-- 7.
c3[0]: 2 <-- 7 , 1.
c3[1]: 6 <-- 0 , 1.
c3[2]: 1 <-- 6 , 8.
c3[3]: 8 <-- 0 , 1.
c3[4]: 9 <-- 2 , 7.

 For synaptic weight :
Global Minima Ratio : 0.994
Local Minima Ratio : 0.0060000000001
Hamming Distance : 0.004287055667306802
Number of relax : 2.186999999999988
computation time : 48.875
```

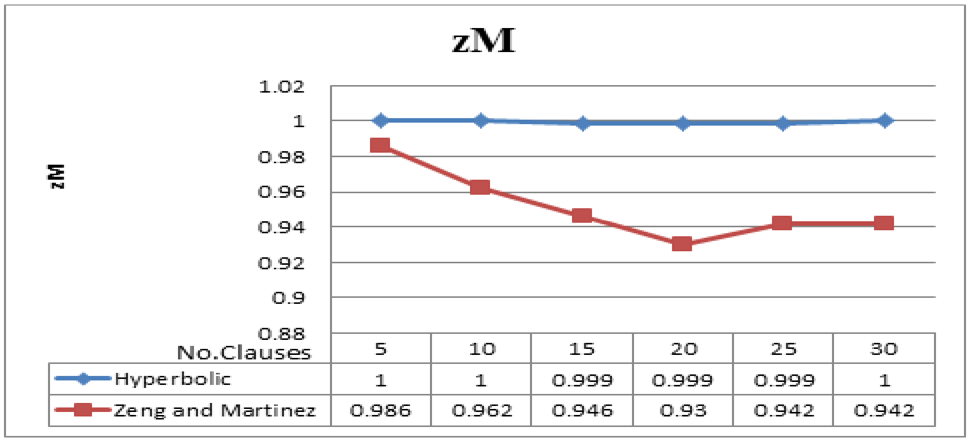
6. Output

Minimum Energy	Number of Relax	Hamming Distance	Global Minima Ratio	Local Minima Ratio
-4.375	2.187	0.0042871	0.994	0.006

Fig. 3. Layout ofAgent Based Modelling using Netlogo

5.1 Ratio of Global Minima (ZM)

We excited the network by using the Zeng function and hyperbolic tangent activation function to accelerate the performance of doing logic programming in Hopfield network.



No.Clauses	5	10	15	20	25	30
Hyperbolic	1	1	0.999	0.999	0.999	1
Zeng and Martinez	0.986	0.962	0.946	0.93	0.942	0.942

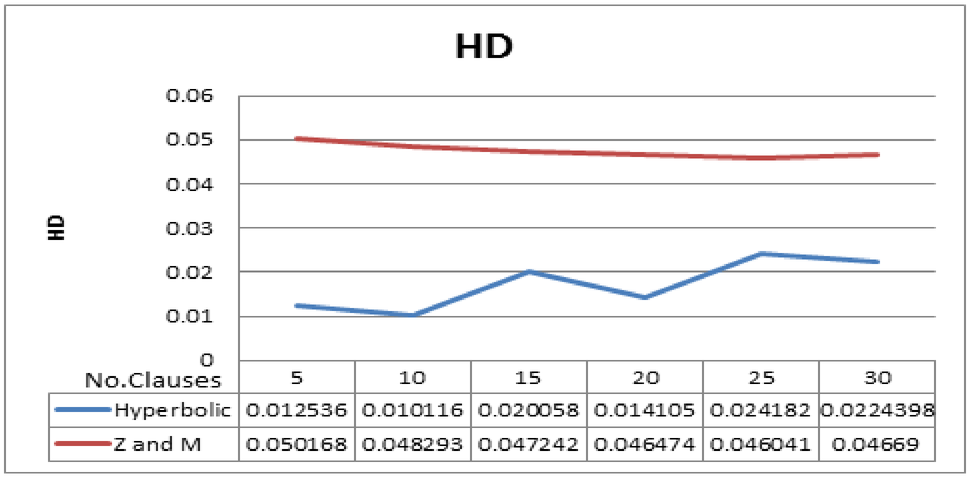
Fig. 4. Global minima ratio for hyperbolic tangent activation function and Zeng function for NC1,2,3=5,10,15,20,25,30 and given NN =50

Figure 4 above shows graphs for ratio of global minima obtained for hyperbolic tangent activation function and Zeng function based on Wan Abdullah's method for different number of literals per clauses respectively at the difference number of neurons. The formula for the ratio of global solution or ratio of global minima was shown in equation (3)

$$\text{Ratio of global minima solutions} = \frac{\text{Number of global solutions}}{\text{Number of trials or iterations}} \qquad (3)$$

It is interesting to note that in Figure 4, we found that the ratio of global solutions for hyperbolic tangent activation function is closer to 1 compared to Zeng function, although we increased the network complexity by increasing the number of neurons (*NN*) and number of literals per clause (*NC1, NC2, NC3*).

5.2 Hamming Distance(HD)



No.Clauses	5	10	15	20	25	30
Hyperbolic	0.012536	0.010116	0.020058	0.014105	0.024182	0.0224398
Z and M	0.050168	0.048293	0.047242	0.046474	0.046041	0.04669

Fig. 5. Hamming Distance for Hyperbolic Tangent activation function and Zeng function for NC1,2,3=5,10,15,20,25,30 and given NN =50

Figure 5 shows the Hamming distance for Hyperbolic tangent activation function and Zeng function for different number of literals per clause respectively simulated. It shows that the value of Hamming distance is approximately zero for all the cases (*NC1*, *NC2* and *NC3*). However, from the graph obtained, we observed a bit of differences that determine the performance of hyperbolic tangent activation function and Zeng function based on Wan Abdullah's method in doing the logic programming. By referring to the Figure 5, it can be observed the hyperbolic tangent activation function, performs better since the Hamming distance are much closer to zero compared with Zeng function although we increased the network complexity by increasing the number of neurons (*NN*) and number of literals per clause (*NC1, NC2, NC3*). This is due to the neurons that already moving to stable states after hyperbolic tangent activation function implemented during the training process. Thus, after the energy relaxing looping, the gap between the global solutions and the states are very close. The more closely the Hamming distance to zero indicates that the performance is superior and vice versa. As the complexity of the network increased, the ability to sustain a huge number of neurons is the main muscle for the hyperbolic tangent activation function.

5.2 Computation Time [CT]

Computation time is an indicator to shows the performance of the activation functions to doing logic programming. Table 4 shows the comparison of computation time between the hyperbolic tangent activation function and Zeng function. According to the table also, hyperbolic tangent activation function resulted in the most successful one compared with Zeng function. Zeng function has lesser ability to doing logic programming compared to hyperbolic tangent activation function. Simulation results show that hyperbolic tangent activation function performs better recognition accuracy than those of the other function.

Table 4: The computation time between hyperbolic tangent activation function and Zeng function (in seconds) for NC1=NC2=NC3= 10 and NN 0f 10 until 80.

CT		
NCi	Hyperbolic	Zeng
5	3153.118	3383.412
10	3305.517	3400.174
15	3662.139	3568.599
20	4189.75	4812.078
25	9029.038	9282.391
30	20022.672	20460.7

6 DISCUSSION

According to the results also, hyperbolic tangent activation function resulted in the most successful one compared with Zeng function. Zeng function has lesser ability to doing logic programming compared to hyperbolic tangent activation function. Having compared to their performances, hyperbolic tangent activation function and Zeng activation function, simulation results show that hyperbolic tangent activation function performs better recognition accuracy than those of the other function, hyperbolic tangent activation function can be used in the vast majority of MLP (Multi Layered Perception) application as a good choice to obtain high accuracy[4].

7 CONCLUSION

Based on the comparison and results obtained (Ratio of Global Minima, Hamming Distance and Computation Time), we can validate the logic program model. This model can be upgraded and accelerated by using hyperbolic tangent activation function and Zeng activation function. According to our experimental study, it is concluded that the hyperbolic tangent activation function can be used in the vast majority of ANN applications as a good choice to obtain high accuracy.

8 REFERENCES

[1] Wan Abdullah, W. A. T. (1991). Neural Network logic. In O. Benhar et al. (Eds.),Neural Networks: From Biology to High Energy Physics. Pisa: ETS Editrice, 135-142.

[2] McCulloch &Pitts .(1943).Schematic diagram of a McCulloch-Pitts neuron model image (Figure 3.1) .[Online],[Accessed 2nd April 2016]. Retrieved from:http://neuralfuzzy.blogspot.com/

[3] Bekir, K. & A Vehbi, O. (2016). Performance Analysis of Various Activation Functionin Generalized MLP Architecture of Neural Networks. International Journal ofArtificial Intelligence and Expert System, (IJAE),Volume(1):1(4), pp.111-122.

[4] Sibi,P. & Jones,S.A. & Siddarth,P. (2013) Analysis of Different Activation Functions using Back Propagation Neural Networks. Journal of Theoretical and Applied Information Technology.

[5] Gourav Garg&Poonam Sharma.(2014). An Analysis of Contrast Enhancement using Activation Functions (Figure 3.2,Figure 3.3&Figure 3.4). International Journal of Hybrid Information Technology Vol.7, No.5 (2014)[Accessed 3rdApril 2016]http://www.sersc.org/journals/IJHIT/vol7_no5_2014/22.pdf

[6] Motcha, M. R. P. &Manimala, K. (2014),Implementation of Hyperbolic Tangent Activation Function in VLSI, International Journal of Advanced Research in Computational Science & Technology, Vol. 2, p 225–228.

[7] Sathasivam, S. & Ng, P.F. (2013). Developing Agent Based Modelling for Doing Logic Programming in Hopfield Network. Applied Mathematical Sciences, Vol. 7, 2013, no. 1, 23 - 35.

[8] Zeng and Martinez , X., & Martinez, T. R. (1999). A new activation function in the Hopfield Network for Solving Optimization Problems. In Artificial Neural Nets and Genetic Algorithms (pp. 73-77). Springer, Vienna.

[9] Sathasivam, S. (2010, December). Usage Of New Activation Function In Neuro-Symbolic Integration. In AIP Conference Proceedings (Vol. 1325, No. 1, pp. 171-174). AIP.

[10] Heckbert, Scott, Tim Baynes, and Andrew Reeson. Agent-based modeling in ecological economics. Annals of the New York Academy of Sciences 1185(1), pp. 39-53, 2010.