International Journal of Computational Bioinformatics and In Silico Modeling
Vol. 6, No. 2 (2017): 911-921
Research Article
Open Access

ISSN: 2320-0634

Analysis of Performance of Various Activation Functions for doing the logic programming in Hopfield Network

Shehab Abdulhabib Saeed Alzaeemi*, Saratha Sathasivam and Salaudeen Abdulwaheed Adebayo

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang Malaysia.

*Corresponding author: Shehab Abdulhabib Saeed Alzaeemi; email: shehab_alzaeemi@yahoo.com

Received: 09 February 2017 Accepted: 27 February 2017 Online: 02 March 2017

ABSTRACT

There are number of common activation functions in use with artificial neural networks (ANNs). The most common choice of activation functions for artificial neural networks (ANNs) is used as transfer functions in research and engineering. The most common reasons for the use of this popularity 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. Objective: The purpose of this paper was to find out the most effective activation function in doing logic programming in the context of Hopfield network. Methods: A comparing hyperbolic tangent activation function, bipolar activation function, unipolar activation function and McCulloch-Pitts function were carried out based on Wan Abdullah's method, evaluations from global minima ratio, Hamming distance and computational time. These functions are used in activation function in a logic program for experimental comparisons. Additionally, computer simulations have been tested using software NETLOGO 5.3.1(64bit) based on Wan Abdullah's method doing the logic programs to demonstrate the ability of Hyperbolic tangent activation function, Unipolar activation function, Bipolar activation function and McCulloch-Pitts function. Hyperbolic tangent activation function resulted in the most successful one compared with bipolar activation function, unipolar activation function, and McCulloch-Pitts function. According to our experimental study, we can say that the hyperbolic tangent activation function can be used in the vast majority of ANN applications as a good choice to obtain high accuracy.

Keywords: Logic Programming in Hopfield network, Types Activation Functions, Implementation of Activation Functions.

1. INTRODUCTION

The first activation function that can be implemented in logic programming within Hopfield neural network is the sign function of McCulloch-Pitts (ideal model) proposed by Wan Abdullah [1]. While McCulloch-Pitts Activation Function may helps the network to find global solution, this function offers a few weaknesses associated with computational burdening and lack of efficiency in producing desired results[2, 14]. However, McCulloch-Pitts neuron can be used in many ways such as the use activation functions other than threshold function are one of the obvious generalizations [2]. The activation function (sometimes called a "transfer) is defined as the output of the neuron by the given input [3,4]. The activation function for the original McCulloch-Pitts neuron is the unit step function. The artificial neuron model, however, has been expanded to include other functions such as the sigmoid, piecewise linear and transfer function. The most common for use of transfer functions are logistic sigmoid function and Gaussian basis functions[5]. The central theme of this dissertation is the study of hyperbolic tangent activation function, unipolar activation function, bipolar activation function and

McCulloch-Pitts function. Detail evaluation (computer simulation) is carried out to compare the effectiveness of these functions. Therefore, the aim of this paper is to compare the performance of the following activation functions: Hyperbolic tangent activation function, bipolar activation function, uni-polar activation function and McCulloch-Pitts activation function.

2. Logic Programming in Hopfield Network

Basic combinatorial relationship between two logic clauses is used to understand Hopfield neural network. Let's assume that *A* and *B* as two clauses,

Table 1: Basic Logic Operators and their Results

A	B	$\neg A$	$\neg B$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$B \leftrightarrow A$	$A \leftrightarrow B$
0	0	1	1	0	0	1	1	1
0	1	1	0	0	1	1	0	0
1	0	0	1	0	1	0	1	0
1	1	0	0	1	1	1	1	1

where $\neg A$ means " NOT *A* "

$A = \neg(\neg A)$
A, B (or $A \wedge B$) means "*A* AND *B*"
$A \vee B$ means "*A* OR *B*"
$A \rightarrow B$ represents "*A* IF *B*"
$A \leftrightarrow B$ represents "*A* IF AND ONLY IF *B*"

According to Wan Abdullah [1,15], *direct method* proposed the best solutions, given the clauses in the logic program, and the corresponding solutions may change as new clauses are added. For example, interpretation with inconsistent logic program can be obtained by using Wan Abdullah's method. Wan Abdullah identified a symmetry mapping between cost functions of neural networks and propositional logic formula. He handles non-monotonicity of logic to model and solve combinatorial optimization problems by using Hopfield network. Wan Abdullah's effort revolves around propositional Horn clauses and learning ability of the Hopfield network.

Based on Wan Abdullah's method, the following algorithm summarizes how a logic program can be implemented in a Hopfield network based on proposal by Wan Abdullah [1] known as direct method[1,15,16]:

(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 $\frac{1}{2}(1+S_x)$ 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 $\frac{1}{2}(1-S_x)$; 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.

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

(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[17,18].

Consider the following example of logic program,

$$P = A \leftarrow B, C \wedge D \leftarrow B \wedge C \quad (2)$$

Given the goal

$$\leftarrow G$$

$A \vee \neg B \vee \neg C$, $D \vee \neg B$ and C are respectively translated from (2) and G is a conjunction of neurons. Since using Wan Abdullah's method, interpretation of inconsistent clauses in the logic program can be obtained, we require to show that $P \wedge \neg G$ is inconsistent in order to prove the goal. Let assign that the values of 1 and -1 are *true* and *false* respectively. Hence $\neg P = -1$ indicates a consistent interpretation while $\neg P = 1$ shows that at least one of the clauses in the program is not satisfied. Translate all clauses and negation in the logic program into Boolean algebraic form[15]:

$$
\begin{aligned}
P &= (A \vee \neg B \vee \neg C) \wedge (D \vee \neg B) \wedge C \\
&= [\neg (B \wedge C) \vee A] \wedge (\neg B \vee D) \wedge C \\
&= (A \vee \neg B \vee \neg C) \wedge (\neg B \vee D) \wedge C \\
\neg P &= \neg [(A \vee \neg B \vee \neg C) \wedge (\neg B \vee D) \wedge C] \\
&= \neg (A \vee \neg B \vee \neg C) \vee \neg (\neg B \vee D) \vee \neg C \\
&= (\neg A \wedge B \wedge C) \vee \neg (B \wedge \neg D) \vee (\neg C)
\end{aligned} \tag{3}
$$

A cost function for bipolar neuron to be minimized can be written as follow:

$$E_p = (1 - V_A)\, V_B\, V_C + (1 - V_D)\, V_B + (1 - V_C) \tag{4}$$

where V_i represent the truth values of $i = A, B, C, D$.

By using step in Wan Abdullah's method where $V_i = \frac{1}{2}(1 + S_i)$,

$$
\begin{aligned}
E_P &= \left[1 - \tfrac{1}{2}(1 + S_A)\right]\left[\tfrac{1}{2}(1 + S_B)\right]\left[\tfrac{1}{2}(1 + S_C)\right] \\
&\quad + \left[1 - \tfrac{1}{2}(1 + S_D)\right]\left[\tfrac{1}{2}(1 + S_B)\right] + \left[1 - \tfrac{1}{2}(1 + S_C)\right] \\
E_P &= \tfrac{1}{2}(1 - S_A)\tfrac{1}{2}(1 + S_B)\tfrac{1}{2}(1 + S_C) + \tfrac{1}{2}(1 - S_D)\tfrac{1}{2}(1 + S_B) + \tfrac{1}{2}(1 - S_C) \\
E_P &= \tfrac{1}{8}(1 - S_A)(1 + S_B)(1 + S_C) + \tfrac{1}{4}(1 - S_D)(1 + S_B) + \tfrac{1}{2}(1 - S_C) \qquad (5)
\end{aligned}
$$

where S_i represent the truth values of $i, i = A, B, C, D$. Notice that we have chosen the multiplication operation to represent the relationship "AND" and addition operation "OR". The minimum value for E_P is 0, corresponding to the fact that all clauses are satisfied. The value for E_P (which is a *n* integer) is proportional to the number of clauses unsatisfied. Table 2 illustrate truth table for $P = \{A \leftarrow B, C, D \leftarrow B, C \leftarrow\}$.

Table 2: The Truth Table, The Number of Unsatisfied Clauses and E_P

S_A	S_B	S_C	S_D	$A \leftarrow B,C$	$D \leftarrow B$	$C \leftarrow$	Num. of unsatisfied clauses	E_P
-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	0	0
-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	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	0	0
1	-1	1	1	1	1	1	0	0
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	0	0

An energy function is defined as follow:

$$E = -\frac{1}{3}\sum_i \sum_j \sum_k J^{(3)}_{[ijk]} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j J^{(2)}_{[i]} S_i S_j - \sum_i J^{(1)}_{[i]} S_i \tag{6}$$

$$E = -\frac{1}{3}\left(6 J^{(3)}_{[ABC]} S_A S_B S_c + 6J^{(3)}_{[ABD]} S_A S_B S_D + 6J^{(3)}_{[ACD]} S_A S_C S_D + 6J^{(3)}_{[BCD]} S_B S_C S_D\right)$$

$$-\frac{1}{2}\left(2J^{(2)}_{[AB]} S_A S_B + 2J^{(2)}_{[AC]} S_A S_C + 2J^{(2)}_{[AD]} S_A S_D + 2J^{(2)}_{[BC]} S_B S_C + 2J^{(2)}_{[BD]} S_B S_D + 2J^{(2)}_{[CD]} S_C S_D\right)$$

$-(J_A^{(1)} S_A + J_B^{(1)} S_B + J_C^{(1)} S_C + J_D^{(1)} S_D)$

where [*ABC*] denote any permutation of *A, B, C.*

The updating rule reads

$$S_i(t+1) = sgn[h_i(t)]$$

where *sgn* is the signum function, (7)

and where local field is given by the equation below.

$$h_i(t) = \sum_j \sum_k J^3_{[ijk]} S_j(t) S_k(t) + \sum_j J^2_{[ij]} S_j(t) + J_i^{(1)} \quad (8)$$

The value of synaptic strengths can be obtained by comparing the cost function E_P and the energy function E.

Table 3: The Clauses and Corresponding Synaptic Strengths (Wan Abdullah Method).

Synaptic Strengths	Clauses			
	$A \leftarrow B,C$	$D \leftarrow B$	$C \leftarrow$	$A \leftarrow B,C \wedge D \leftarrow B \wedge C \leftarrow .$
$J^{(3)}_{[ABC]}$	1/16	0	0	1/16
$J^{(3)}_{[ABD]}$	0	0	0	0
$J^{(3)}_{[ACD]}$	0	0	0	0
$J^{(3)}_{[BCD]}$	0	0	0	0
$J^{(2)}_{[AB]}$	1/8	0	0	1/8
$J^{(2)}_{[AC]}$	1/8	0	0	1/8
$J^{(2)}_{[AD]}$	0	0	0	0
$J^{(2)}_{[BC]}$	-1/8	0	0	-1/8
$J^{(2)}_{[BD]}$	0	1/4	0	1/4
$J^{(2)}_{[CD]}$	0	0	0	0
$J^{(1)}_{[A]}$	1/8	0	0	1/8
$J^{(1)}_{[B]}$	-1/8	-1/4	0	-3/8
$J^{(1)}_{[C]}$	-1/8	0	1/2	3/8
$J^{(1)}_{[D]}$	0	1/4	0	1/4

By substituting the values of connection strengths into the energy function, E in equation (6), the global minimum energy can be obtained. So, the global minimum energy can be specified as

$$E_{min} = -\frac{1}{3}\left(6\, J^{(3)}_{[ABC]} S_A S_B S_c + 6J^{(3)}_{[ABD]} S_A S_B S_D + 6J^{(3)}_{[ACD]} S_A S_C S_D + 6J^{(3)}_{[BCD]} S_B S_C S_D\right)$$

$$-\frac{1}{2}\left(2J^{(2)}_{[AB]} S_A S_B + 2J^{(2)}_{[AC]} S_A S_C + 2J^{(2)}_{[AD]} S_A S_D + 2J^{(2)}_{[BC]} S_B S_C + 2J^{(2)}_{[BD]} S_B S_D + 2J^{(2)}_{[CD]} S_C S_D\right)$$

$$-(J_A^{(1)} S_A + J_B^{(1)} S_B + J_C^{(1)} S_C + J_D^{(1)} S_D)$$

$S_A = S_B = S_C = S_D = 1$ Substitute into equation above,

$$E_{min} = -\frac{7}{8}$$

Hence, the global minimum energy for this example is $-\frac{7}{8}$.

Conventionally, we used [-1,1] to represents the [truth, false] values respectively. Hopfield proposed a similarity to Ising spins, where the values for the spin can be taken as 0 or 1, also known as [0,1] convention. Hence, the new assignment is the values of 1 defined as *true* and 0 defined as *false.* Therefore, $\neg P = 0$ indicates a consistent interpretation otherwise $\neg P = 1$ indicates unsatisfied clauses in the program.

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. The authors[3,4] also suggested that the operation of an artificial neural network is to sum up the product associated with weight and the input signal and produce an output or activation function. The most common for the use of activation functions are: McCulloch-Pitts function, hyperbolic tangent activation function, bipolar activation function and Unipolar activation function.

3.1 Uni-Polar Sigmoid Function

The Unipolar Function was tested by using logarithmic function where the input has any value between -∞ and ∞ and the output is limited between values of 0 and 1.

The logarithmic sigmoid function is given by:

$$g(x) = \frac{1}{1 + e^{-x}} \quad (9)$$

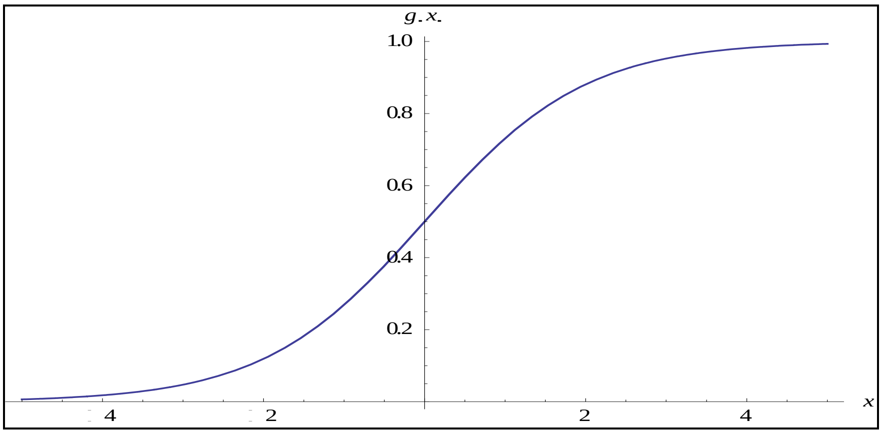

Figure 1: Uni-Polar Function [6]

Note: The application has been utilized in ANN to predict stock market indices [7].

3.2 Bipolar Sigmoid Function

the bipolar function is a similar to the sigmoid function but this activation function takes the input which may have any value between -∞ and ∞ and the output is changed into -1 to 1.

The function is given by:

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (10)$$

3.3 Hyperbolic Tangent Function

The hyperbolic tangent activation function is the most common activation function for neural network. The function's 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[13].

The function is given by:

$$g(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (11)$$

3.4 McCulloch-Pitts Function

McCulloch-Pitts function is unbounded function based on the total weights and input. Output range is between values of -∞ and ∞.

The function is given by:

$$g(x) = \sum_{i=0}^{n} W_i * x_i \quad (12)$$

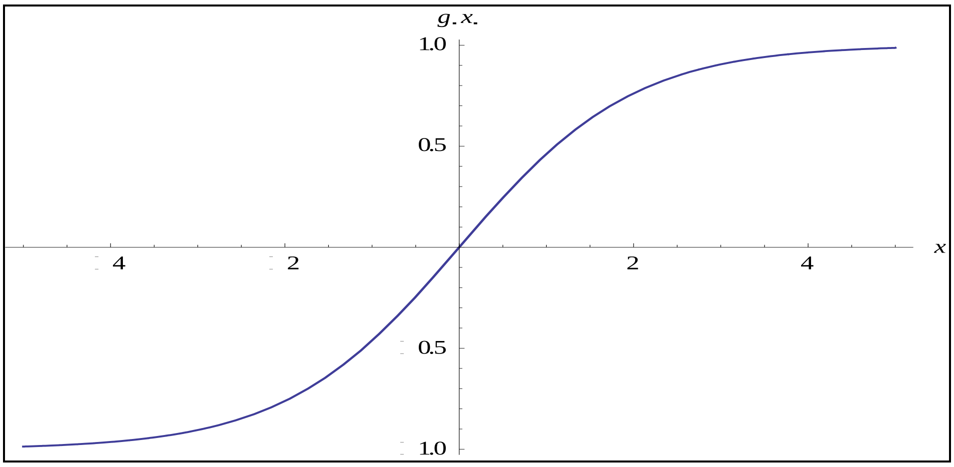

Figure 2: Bi-Polar Function [6]

Note: The application is in multispectral land sat TM image classification to classify the satellite image [21].

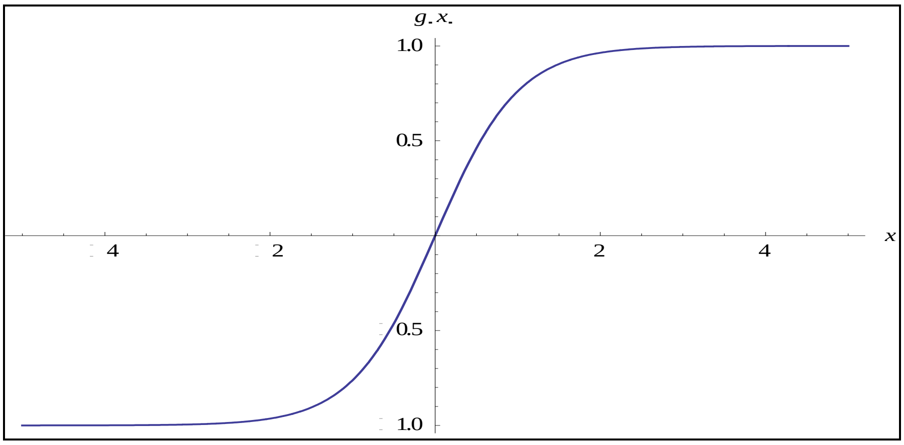

Figure 3: Hyperbolic Tangent Function [6]

Note: The application has been utilized in ANN to classify the breast cancer cells [8,20].

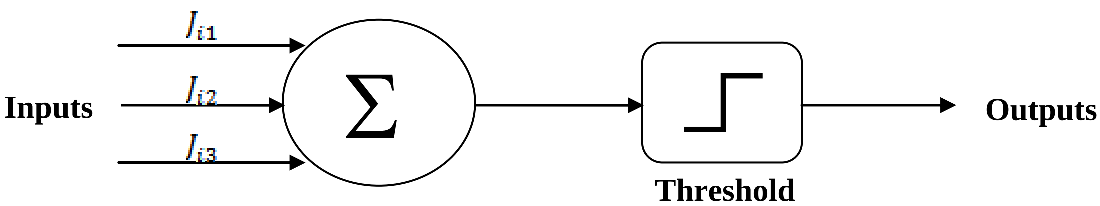

Figure 4: Schematic diagram of a McCulloch-Pitts neuron model [9].

Note: The application has made an Important Contribution to the development of ANN- model key features of biological neurons [10].

4. Implementation of Hyperbolic Tangent Activation Function, Unipolar Activation Function and Bipolar Activation Function in Hopfield Network

The following pseudo code shows how hyperbolic tangent activation function, unipolar activation function and bipolar activation function have been applied in Hopfield neural network.

INPUT:

Number of neurons (*NN*), number of first order clauses (*NC1*), number of second order clauses (*NC2*), number of third clauses (*NC3*), number of Hebbian learning (NH), relaxation time (RELAX), number of trial (NT), maximum combination for neurons (COMBMAX), tolerance value (TOL).

OUTPUT:
Global minima ratio (zM), Hamming distance (HD), computation time (s)

START

1) Set i= 0, 1, 2, ...,NN,
   Energy parameters = 0
   Others parameters = 0
2) Generate minimum energy
   For (i = 0, i<NCz, i++)
   where z = 1,2,3
   Update minimum energy via formula in the equation (13):

$$E_p = \frac{1}{8}(1 - S_A)(1 + S_B)(1 + S_C) + \frac{1}{4}(1 - S_D)(1 + S_B) + \frac{1}{2}(1 - S_C) \quad (13)$$

3) Generate random clauses and synaptic strength
   For (i = 0, i<NCz, i++)
   wherez = 1,2,3
   Update random clause z,
   Set synaptic strength = 0
4) Activation functions

   Define hyperbolic tangent activation function using equation (14)

$$g(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad (14)$$

   or unipolar activation function using equation (15)

$$g(x) = \frac{1}{1 + e^{-x}} \quad (15)$$

   or bipolar activation function using equation (16)

$$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (16)$$

   where $x = \sum_{i=0}^{NN} W_i x_i$

   if (cost function > 0)
   Update synaptic strength
   Calculate total synaptic strength
5) Check final state
   For (m = 0, m<NN, m++)
   Update Final State
6) If (Update Final State = Stable state)
   PROCEED to 7
   else
   RETURN to 3
7) Calculate final energy using equation (17)

$$E = -\frac{1}{3}\sum_i \sum_j \sum_k J^{(3)}_{[ijk]} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j J^{(2)}_{[ij]} S_i S_j - \sum_i J^{(1)}_i S_i \quad (17)$$

   fori, j, k = 1, 2, ... , NN
8) Determine global solution or local minima solution
   If $|E_p - E| \leq 0.001$ (18)
   Final energy = Global solution
   else
   RETURN to (Local solution)
9) OUTPUT (Global minima ratio and Hamming distance)
   END

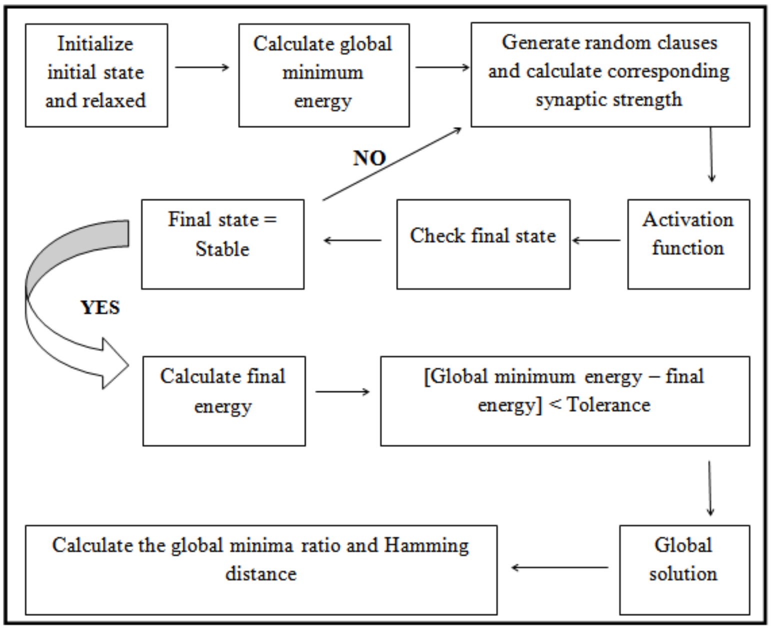

Figure 5: Flow chart on the implementation of hyperbolic tangent activation function, unipolar activation function and bipolar activation function.

4. Experimental Results

As a first step in this process, we create random program clauses, and then we initialize initial states for the neurons in the clauses. After that, the network is left to evolve until it reaches the minimum of the energy. During this step, the neuron state is updated using one of the following functions such as hyperbolic tangent activation function, bipolar activation function, unipolar activation function or McCulloch-Pitts function depend on the user intend to start with. Consequently, we then obtained the ratio of global minima, the hamming distance and computational time (seconds). The three parts were analyzed:

1. Global Minima Ratio

$$\text{Ratio of global minima solutions} = \frac{\text{Number of global solutions}}{\text{Number of trials or iterations}} \quad (3.1)$$

2. Hamming Distance, which is a distance measure between stable state and global solution.
3. Computational time (the time taken for computer or CPU produce results).

We ran the relaxation for 100 trials and 100 combinations of neurons with the given tolerance, 0.001 in order to reduce the statistical error by using attempt and error technique. The procedures are generic and can be used with any activation function or sign function. More specifically, we made comparison according to the Global Minima Ratio, Hamming Distance and Computation Time between the McCulloch-Pitts function, unipolar activation function, bipolar activation function and hyperbolic tangent activation function based on Wan Abdullah's method.

5. Results in NETLOGO

The computer simulations were tested by using NETLOGO[19] software version 5.3.1 which was characterized modern tool and buttons which reduces the length of the program was used essential in getting the results. The global minima ratio, Hamming distance and computation time for the hyperbolic tangent activation function, unipolar activation function, bipolar activation function and McCulloch-Pitts function were obtained 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 validating the performance of each of the hyperbolic tangent activation function, unipolar activation function, bipolar activation function and McCulloch-Pitts function in doing logic program in Hopfield network.

5.1 Ratio of Global Minima (zM)

We stimulated the network by using the four methods individually, McCulloch-Pitts function, unipolar activation function, bipolar activation function and hyperbolic tangent activation function to accelerate the performance of doing logic programming in Hopfield network.

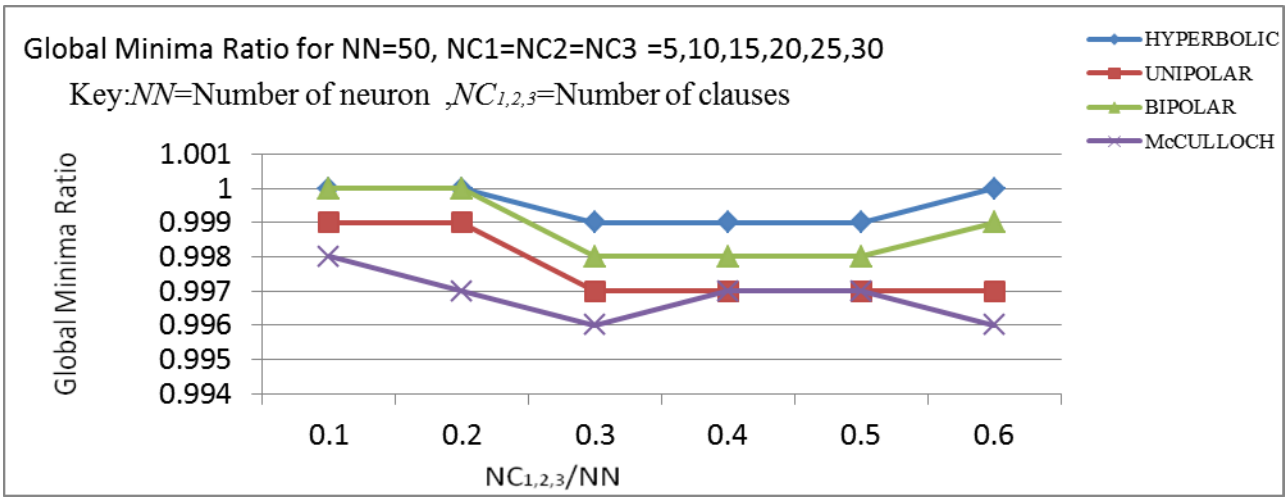

Figure 6: Global minima ratio for hyperbolic tangent activation function, unipolar activation function, bipolar activation function and McCulloch-Pitts function for *NC1,2,3=5,10,15,20,25,30 and* given *NN* =50

Figure 5 above shows graphs for ratio of global minima obtained for hyperbolic tangent activation function, unipolar activation function, bipolar activation function and McCulloch-Pitts 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 (3.1)

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

It is interesting to note that in Figure 5, we found that the ratio of global solutions for hyperbolic tangent activation function is closer to 1 compared to the bipolar activation function, unipolar activation function and McCulloch-Pitts function, although we increased the network complexity by increasing the number of neurons (*NN*) and number of literals per clause (*NC1, NC2, NC3*). Next, bipolar activation function shows better performance compared to the unipolar activation function and McCulloch-Pitts function. Uni-polar activation function, however, shows better performance compared to the McCulloch-Pitts function.

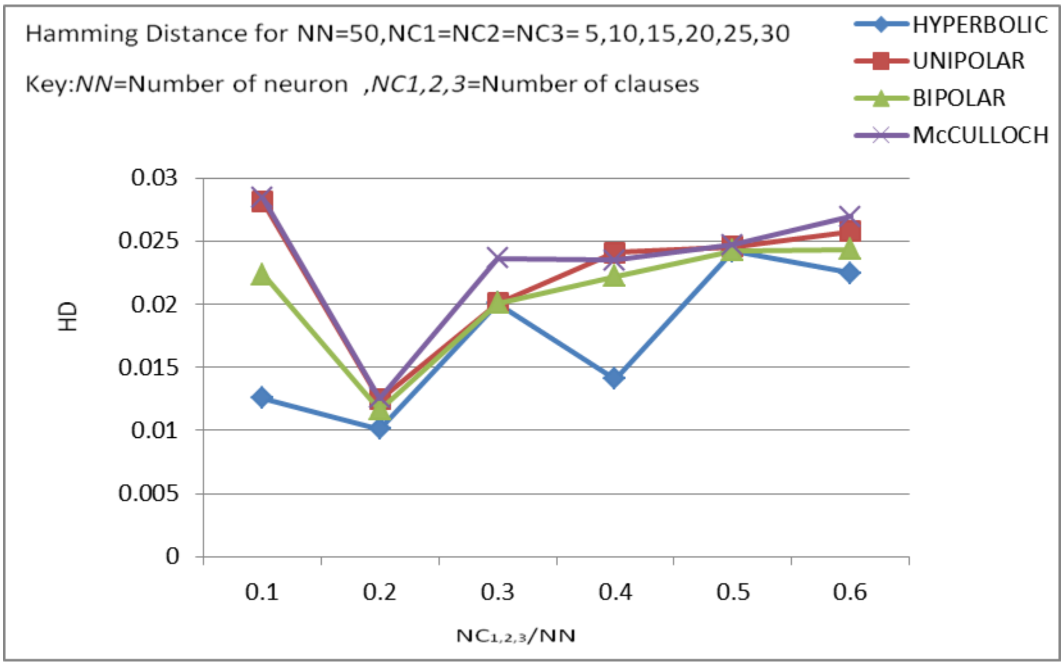

Figure 7: Hamming Distance for hyperbolic tangent activation function, unipolar activation function, bipolar activation function and McCulloch-Pitts functionfor*NC1,2,3=5,10,15,20,25,30 and* given *NN*=50 for NC given NN = 40

5.2 Hamming Distance

Figure 6 shows the Hamming distance for Hyperbolic tangent activation function, Uni-polar activation function, Bipolar activation function and McCulloch-Pitts 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, bipolar activation function, unipolar activation function and McCulloch-Pitts function based on Wan Abdullah's method in doing the logic programming. By referring to the Figure 6, it can be observed the hyperbolic tangent activation function, performs better since the Hamming distance are much closer to zero compared with unipolar activation function, bipolar activation function and McCulloch-Pitts 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. The network will be more complex if the number of neurons increased.

5.3 Computation Time

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, unipolar activation function, bipolar activation function and McCulloch-Pitts function. According to the table also, hyperbolic tangent activation function resulted in the most successful one compared with bipolar activation function, unipolar activation function and McCulloch-Pitts function. McCulloch-Pitts function has lesser ability to doing logic programming compared to hyperbolic tangent activation function. Having compared to their performances, hyperbolic tangent activation function, bipolar activation function and unipolar 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, unipolar activation function, bipolar activation function and McCulloch-Pitts function (in seconds) for *NC1=NC2=NC3*= 10 and NN 0f 10 until 80.

	NC1=NC2=NC3=10			
No. of neurons NN	**Computational time for different Activation functions** (in seconds)			
	Hyperbolic Tangent activation function	Unipolar activation function	Bipolar activation function	McCulloch-Pitts function
10	24.539	49.296	45.224	50.966
20	178.028	350.907	330.798	365.431
30	556.207	1161.016	1093.203	1204.556
40	1360.741	2556.451	2458.222	2634.4
50	2605.517	4469.306	3380.32	4732.996
60	8014.183	9191.458	9014.253	9143.397
70	10523	12973.242	11811.98	13578.677
80	17188.14	18331.697	18287.27	19593.567

6. Discussion

According to the results also, hyperbolic tangent activation function resulted in the most successful one compared with bipolar activation function, unipolar activation function and McCulloch-Pitts function. Having compared to their performances, bipolar activation function and unipolar activation function show less accuracy in order to support more data [11]. McCulloch-Pitts function has lesser ability to doing logic programming compared to hyperbolic tangent activation function. This has been proven by Sunderajoo [12]. Having compared to their performances, hyperbolic tangent activation function, bipolar activation function and unipolar 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 results obtained (Ratio of Global Minima, Hamming Distance and Computation Time), we can validate that the logic program model can be upgraded and accelerated by using hyperbolic tangent activation function, bipolar activation function and unipolar activation function. In this study hyperbolic tangent activation function is more effective compared to bipolar activation function, unipolar activation function and McCulloch-Pitts function due to its effectiveness in doing logic programming in Hopfield network in logic programming. In this study, we have four conventional differentiable and monotonic activation functions for comparison results obtained (Ratio of global minima, hamming distance and computational time). These proposed well-known and effective functions are hyperbolic tangent activation function, bipolar activation function, unipolar activation function and McCulloch-Pitts function. Having compared their performances, simulation results show that hyperbolic tangent (*Tanh*) function performs better recognition accuracy than those of the other functions. This result demonstrates that is possible to improve the ANN performance through the use of much effective activation function. According to our experimental study, we can say 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. (2010). Performance Analysis of Various Activation Functionin Generalized MLP Architecture of Neural Networks. International *Journal of Artificial Intelligence and Expert System.* **1**(4), 111-122.
4. 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), 111-122.
5. 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*.
6. 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

7. Kimoto, T., Asakawa, K., Yoda, M. and Takeoka, M., 1990, June.Stock market prediction system with modular neural networks. In *Neural Networks, 1990., 1990 IJCNN International Joint Conference on* (pp. 1-6). IEEE.
8. Khan, J., Wei, J.S., Ringner, M., Saal, L.H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C.R., Peterson, C. and Meltzer, P.S., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, *7*(6), pp.673-679.
9. Zurada, J. (1992). *Intoduction to Neural Network*. New York: West Publishing Company.
10. Karmiloff-Smith, A.,1998. Development itself is the key to understanding developmental disorders. *Trends in cognitive sciences*, *2*(10), pp.389-398.
11. Karlik, B. & Olgac, A. V. (2010). Performance Analysis of Various Activation Function in Generalized MLP Architecture of Neural Networks. *International Journal of Artificial Intelligence and Expert System*. 1(4), p. 111-122.
12. Sunderajoo, T. (2013).New Algorithm in Training Neural Networks.Msc.Thesis, Universiti Sains Malaysia
13. 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.
14. Piccinini, G. (2004) The first computational theory of mind and brain: a close look at McCulloch and Pitts. *Journal of Synthese*.[Online], [Accessed 9th March 2016]
15. Sathasivam, S. (2006). Logic Mining in Neural Networks. PhD. Thesis. University of Malaya, Malaysia.
16. Sathasivam, S. & Wan Abdullah, W.A.T. (2008). Logic Learning in the Hopfield Networks, *Modern Applied Science,* **2**(2), p. 57-62.
17. Sathasivam, S. (2009), Energy Relaxation for Hopfield Network With the New Learning Rule*. International Conference on Power and Optimazation*, Bali.
18. Sathasivam, S. (2011).Boltzmann Machine and New Activation Function Comparison.*Applied Mathematical Science.***5**(78), p. 3853-3860.
19. 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*.
20. Isa, I.S., Saad, Z., Omar, S., Ahmad, K.A., Sakim, H. A. M. (2010), Suitable MLP Network Activation Functions for Breast Cancer and Thyroid Disease Detection. Proceeding of 2th International on Computational Intelligence, Modelling and Simulation.
21. Solberg, A.H.S., Jain, A.K. and Taxt, T., 1994. Multisource classification of remotely sensed data: fusion of Landsat TM and SAR images. *IEEE transactions on Geoscience and Remote Sensing*, *32*(4), pp.768-778.

*****