



# Boltzmann Machine and Hyperbolic activation function in Higher Order Neuro Symbolic Integration

MuralyVelavan<sup>1</sup>,  
[dsmuraly@yahoo.com](mailto:dsmuraly@yahoo.com)

Zainor Ridzuan bin Yahya<sup>1</sup>,

Mohamad Nazri bin Abdul Halif<sup>2</sup>,

Saratha Sathasivam<sup>3</sup>

<sup>1</sup>Institute of Engineering  
Mathematics, Universiti  
Malaysia Perlis, 02600 Arau,  
Perlis, Malaysia.

<sup>2</sup>School of Microelectronic  
Engineering, Universiti  
Malaysia Perlis, 02600 Arau,  
Perlis, Malaysia.

<sup>3</sup>School of Mathematical  
Sciences, Universiti Sains  
Malaysia Penang, 11800 USM,  
Malaysia

**Abstract** - Higher-order network structure is important in doing higher order programming because high-order neural networks have converge faster and have a higher memory and story capacity. Furthermore higher order networks also have higher approximation ability and robust if compare lower-order neural networks. Thus, the higher-order clauses for logic programming in Hopfield Networks are been focused in this paper. We will limit till fifth order network due to complexity issue. Hereby we employed Boltzmann Machines and hyperbolic tangent activation function to increased the performance of neuro symbolic integration. We used agent based modelling to model this problem.

**Index terms** - Boltzmann machine, agent based modelling and hyperbolic tangent activation function

## I. INTRODUCTION

The field of neuro-symbolic integration is stimulated by the fact that formal theories (as studied in mathematical logic and used in automated reasoning) are commonly recognised as deductive systems which lack such properties of human reasoning as adaptation, learning and self-organisation. On the other hand, neural networks, introduced as a mathematical model of neurons in the human brain, are claimed to possess all of the mentioned abilities, and moreover, they provide parallel computations

and hence can perform certain calculations faster than classical algorithms. Hopfield network is a recurrent neural network [1] invented by John Hopfield, consists of a set of  $N$  interconnected neurons which all neurons are connected to all others in both directions. It has synaptic connection pattern which involving Lyapunov function  $E$  (energy function) for dynamic activities. It serves as content addressable memory systems with binary threshold units. The more explanation will carry out in section 2.

Logic is deals with true and false while in the logic programming, a set of Horn clauses that formed by atoms are represented to find the truth values of the atoms in the clauses. It is using neurons to store the truth value of atoms to write a cost function for minimization when all the clauses are satisfied. In addition, a bi-directional mapping between propositional logic formulas and energy functions of symmetric neural networks had defined by Gadi Pinkas [2, 3] and Wan Abdullah [4, 5]. Further detail can refer to the references. The advantages by using Wan Abdullah's method are it can revolves around propositional Horn

Technical Article

First Online on – 30 March 2015, Revised on – 30 June 2020

© 2020 RAME Publishers

This is an open access article under the CC BY 4.0 International License

<https://creativecommons.org/licenses/by/4.0/>

**Cite this article** – MuralyVelavan, Zainor Ridzuan bin Yahya, Mohamad Nazri bin Abdul Halif, Saratha Sathasivam, "Boltzmann Machine and Hyperbolic activation function in Higher Order Neuro Symbolic Integration", *International Journal of Computational and Electronics Aspects in Engineering*, RAME Publishers, vol. 1, issue 2, pp. 63-69, 2015, Revised in 2020.

<https://doi.org/10.26706/ijceae.1.2.20150103>

clauses and learning ability of the Hopfield network and hunts for the best solutions, given the clauses in the logic program, and the corresponding solutions may change as new clauses added.

In this paper, we had developed agent based modelling (ABM) for integration of Boltzmann machine and hyperbolic activation function in higher Order Hopfield Networks. An ABM is a new computational modelling paradigm which is an analysing systems that representing the ‘agents’ that involving and simulating of their interactions. Their attributes and behaviours will be group together through their interactions to become a scale.

This paper is organized as follows. In section 2, an outline of Hopfield network is given and in section 3, method of doing logic programming in neural network is described. In section 4, agent based modelling will be discussed. Meanwhile in section 5 contain discussions regarding the Boltzmann machine and Hyperbolic Tangent activation function. Finally, section 6 and 7 occupy the simulation results and concluding remarks regarding this work.

## II. HIGHER ORDER HOPFIELD NETWORKS

Discrete Hopfield network [6] is shown in Figure 1, is an expanded form of a common representation of the Hopfield network. Hopfield had stated that this network is useful for solving combinatorial optimization problems as a content addressable memory or an analog computer. Combinatorial optimization includes looking for the combination of choices from a discrete set which produces an optimum value for some related cost function. In neural network, higher order logic programming [7,8] is highly regarded as the essential method in Hopfield Networks. When the Hopfield neural network is used to solve NP-complete optimization problem such as Travelling salesman problem, positive solutions would be produced. In our work we classify forth and fifth order network as higher order networks. While in [8], the convergence property had restudied before put in application in real life.

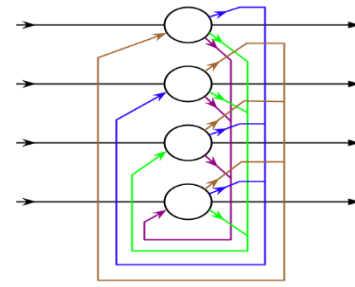


Figure 1: Discrete Hopfield network with four neurons

The most important paper that acts as the main backbone of this paper is [8]. It is a study of the different dynamics in HOHN and problem affecting practical application of these networks are brought to light such as incoherence between the network dynamics and the associated energy function, error due to discrete simulation on a digital computer, existence of local minima and convergence depends on coefficients weighting the cost function terms. From those stated above, Hopfield network has overcome the difficulty to find suitable parameters to guarantee convergence. Beside, HOHN had applied in data mining such as forecasting and simulating data. It can reduce network size, improved simulation errors. Other than that, Higher-order Hopfield Networks can solve non-linear and discontinuous data in larger field and connections. For example, it is well performed in nonlinear statistical modelling and it can provide a new alternative to logistic regression in bigger state and numbers. Furthermore, it is able to detect all possible interactions between predictor variables such as detect complex nonlinear relationships between dependent and independent variables. Lastly, it can be used as research tool like neurobiologists use it for interpretation of neurobiological phenomena. From here, researchers know that Hopfield network can use to minimize a configurationally energy function and thus can solve the combinatorial optimisation problem. It is a reason why there are good solutions can be found. So, in order to make the network more robust we present higher order connections.

The Higher-order Hopfield Networks with the order =  $n-1$  is stated as below. The energy function is,

$$E = -\frac{1}{n} \sum_{i_1} \sum_{i_2} \sum_{i_3} \dots \sum_{i_n} w_{i_1 i_2 i_3 \dots i_n} x_{i_1} x_{i_2} x_{i_3} \dots x_{i_n} -$$

$$\frac{1}{n-1} \sum_{i_1} \sum_{i_2} \sum_{i_3} \dots \sum_{i_{n-1}} w_{i_1 i_2 i_3 \dots i_{n-1}} x_{i_1} x_{i_2} x_{i_3} \dots x_{i_{n-1}} -$$

$$\dots -$$

$$\frac{1}{2} \sum_{i_1} \sum_{i_2} w_{i_1 i_2} x_{i_1} x_{i_2} - \sum_i w_i x_i \tag{1}$$

where  $x_i$  is the state value of *ith* neuron. The weights defines the connection weights of the *n*th order connection from neurons  $i_1 i_2 i_3 \dots i_n$  to neuron *I*,  $h_i$  is the input potential to neuron *I* and  $x_i$  is the state of neuron *i*. In the high-order model each node is assigned a sigma-pi unit that updates its activation value by first computing the partial derivative of the energy function. The dynamic equation or the updating rule of the network is

$$x_i(t) = \text{sgn}(h_i(t)),$$

$$h_i = \sum_{i_1} \sum_{i_2} \sum_{i_3} \dots \sum_{i_n} w_{i_1 i_2 i_3 \dots i_n} x_{i_1} x_{i_2} x_{i_3} \dots x_{i_n} +$$

$$\sum_{i_1} \sum_{i_2} \sum_{i_3} \dots \sum_{i_{n-1}} w_{i_1 i_2 i_3 \dots i_{n-1}} x_{i_1} x_{i_2} x_{i_3} \dots x_{i_{n-1}} + \dots +$$

$$\sum_{i_1} \sum_{i_2} w_{i_1 i_2} x_{i_1} x_{i_2} +$$

$$\sum_i w_i x_i \tag{2}$$

where *sgn* is signum function. The connection weight of higher-order Hopfield networks is symmetrical.

As the dynamic equation is derived by partial derivative, it only guarantees convergence towards local minimum. It is affected by the value of the coefficients that weight the different terms of the cost function [9, 10, 11]. Thus, the Boltzmann machine and Hyperbolic Tangent activation function will carry out in this paper. Further information will explain in section 4.

### III. LOGIC PROGRAMMING

Logic programming is the use of mathematical logic for computer programming. Thus, higher order Hopfield network had carried out in logic programming model. A HOHN is used to minimise logical inconsistency in interpretations of logic programs and clauses. To apply it, first of all need to understand what logic program play in

the system. Following is the logic program that built by using Wan Abdullah's method in HN.

Following is the algorithms.

- i) Given a logic program, translate all the clauses in the logic program into basic Boolean algebraic form. It like  $A \leftarrow B, C$  as  $A \vee \neg (B \wedge C) = A \vee \neg B \vee \neg C$
- ii) Identify a neuron to each ground neuron.
- iii) Initialize all connections strengths to zero. It assumed the connection with A, B and C is zero value.
- iv) Derive a cost function that is associated with the negation of all the clauses, such that  $\frac{1}{2}(1 + S_A)$  represents the logical value of a neuron *A*, where  $S_A$  is the neuron corresponding to *A*. The value of  $S_A$  is define in such a way that it carries the values of 1 if *A* is true and -1 if *A* is false. Negation (neuron *A* does not occur) is represented by,  $\frac{1}{2}(1 - S_A)$ ;  $E_p = \frac{1}{2}(1 - S_A) \frac{1}{2}(1 + S_B) \frac{1}{2}(1 + S_C) + \dots$ ...a conjunction logical connective 'and' is represented by multiplication whereas a disjunction connective 'or' is represented by addition.
- v) Obtain the values of connection strengths by comparing the cost function with the energy, *H* which in the section 2 that had recognized in Hopfield network.
- vi) Let the neural networks evolve until minimum energy is reached. The neural states then provide a solution interpretation for the logic program, and the truth of ground atom may be checked then consider the solution obtained is a global solution or not.

A logic program contains of program clauses and it is activated by an initial goal. It is easy to understand, modify and verify. For example in a simple propositional case, logic clauses had formed as  $A \leftarrow B_1, B_2, B_3, \dots, B_n$  where the arrow can be read as 'if' while the comma can be read as 'and' for the purpose of interpretation the clauses by using truth value. Thus, a model or pattern can be found to the given logic program and it can be a way to solve the combinational optimization problem. Consequently, to carry out a logic program, we need to build up a simulator to run it. However, to solve the global minima problem, in next section, an introduction about Boltzmann machine and Hyperbolic Tangent activation function will carry out.

#### IV. BOLTZMANN MACHINE

A Boltzmann machine, like a Hopfield network, is a network of units with an “energy” defined for the network. Unlike Hopfield networks, binary units of Boltzmann Machine are stochastic. By referring to the energy function,  $E(u)$  for Hopfield networks, due to the probabilistic update rule, a Boltzmann machine is able to transit the states on higher energy level in contrast to Hopfield network. This feature can avoid the network getting stuck in local minima of the energy function in minimization problems [12]. The difference in the global energy that results from a single unit  $i$  being 0 versus 1, written  $\Delta E_i$ , is given by:

$$\Delta E_i = \sum_{j=1}^n u_j w_{ij} - \theta_i \quad (3)$$

Thus,  $\Delta E_i > 0$  means that the energy of the whole system is higher if  $u_i = 0$  and higher energy for  $\Delta E_i < 0$  with  $u_i = 1$ . A Boltzmann machine is made up of stochastic units. The probability,  $p_i$  of the  $i$ -th unit being on is given by:

$$p_i = \frac{1}{1 + e^{\left(\frac{-\Delta E_i}{T}\right)}} \quad (4)$$

where the scalar  $T$  is referred to as the temperature of the system. The network is run by repetitively choosing a unit and setting its state. After running for long period at a certain temperature, the probability of a global state of the network will depend only upon that global state’s energy based on Boltzmann distribution. It is true when the probability distribution of global states has converged. A temperature value is needed to influence the output from the network. As this temperature decreases from a high temperature to reach a thermal equilibrium at a low temperature, we are guaranteed to converge to a distribution where the energy level fluctuated around the global minimum. This process is called simulated annealing. This material is subjected to high temperature and then gradually cooled. The gradual cooling allows the material to cool to a state in which there are few weak point, the molecule system becomes more and more stable. The heat causes the atoms to become unstuck from their initial position (local minimum) and move randomly to the states of higher energy. It achieves global optimum

whereas the entire object achieves a minimum energy crystalline structure.

#### A. Hyperbolic Tangent Activation Function In Hopfield Network

Hyperbolic tangent activation function is one of well know activation function. However, logistic function which was frequently in use in neural network, introduced by McCulloch-Pitts where it is already established in original method of doing logic programming in Hopfield network proposed by Wan Abdullah. Since McCulloch-Pitts function is unbounded, smoother function such as hyperbolic tangent which have output range between -1 and +1 are preferred. In this section, we will discuss the application and implementation of Hyperbolic Tangent activation function.

Most neurons in neural networks using a scalar-to-scalar function which was called activation function to transform their net input while the activation value is fed via synaptic connections to one or more other neurons. The activation function is sometimes called a transfer function. The transfer function with a bounded range are frequently called squashing functions. Hyperbolic tangent activation function is one of an example of squashing function.

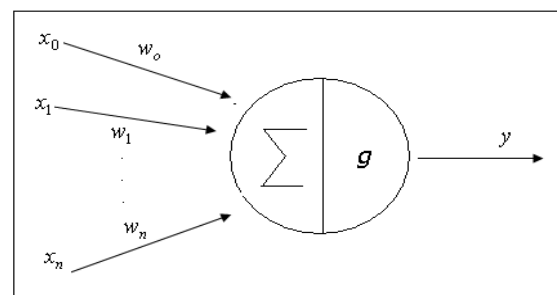


Figure 2: An artificial neuron with activation function

Figure 2 shows the artificial neuron  $x$  which is a neuron with  $n$  input dendrites ( $x_0 \dots x_n$ ) and where ( $w_0 \dots w_n$ ) are weights of inputs and one output axon  $y(x)$ .  $g$  is an activation function that weights how dominant the output should be from the neuron based on the sum of the input. Equation 3.6 shows the equation of activation function.

$$y(x) = g\left(\sum_{i=0}^n w_i x_i\right) \quad (5)$$

Where  $g(x)$  used the hyperbolic tangent function as below,

$$g(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{6}$$

At first, we substitute the total of weights and inputs into Hyperbolic Tangent function to achieve the value. If it surpasses the threshold ( $\theta = 0$ ), the actual output will be 1 else the value of the actual output is -1. After obtained an output pattern during the network computation, if error exists, or a difference between actual and desired output patterns, the weights will be adjusted to reduce the error.

### V. AGENT BASED MODELLING

Firstly, a simulator of Hopfield networks that using a conventional computer had created instead of every time build up a new network design or store a new set of memories. We used NETLOGO version 6.0 as the platform. It saves lots of energies and times for the programmer to rebuild new system from time to time. Thus, a computer program which emulates exactly what the user want needs to construct in order to simulate the action of Hopfield Network. It will be easier for the programmer to modify the program and store a new set of data. Thus, an agent based modelling had designed for the user to run the simulator. In this paper, an agent based modelling which was implemented the Higher Order Hopfield Network (HOHN) had created.

Moreover, agent-based Modelling (ABM) which also called individual-based modelling is a new computational modelling paradigm which is an analyzing systems that representing the ‘agents’ that involving and simulating of their interactions. Their attributes and behaviours will be group together through their interactions to become a scale. Programmer can design ABM in Netlogo by using button, input, output, slides and other functions that make ABM easy to understand and use. In addition, ABM reveals the appearance of the systems from low to high level outcomes and it make improvement by surpassing the traditional modelling limitations such as allowing agent learning and adaption, limited knowledge and access to information. So, by using this approach we can get a clear visualization on doing logic programming in HOHN. The following

screenshots show the layout of the ABM for boltzmann machine and hyperbolic activation function.

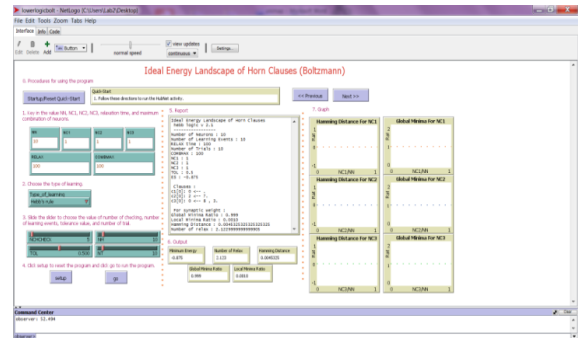


Figure 3: Agent Based Modelling for integration of Boltzmann Machine in logic programming of HONN

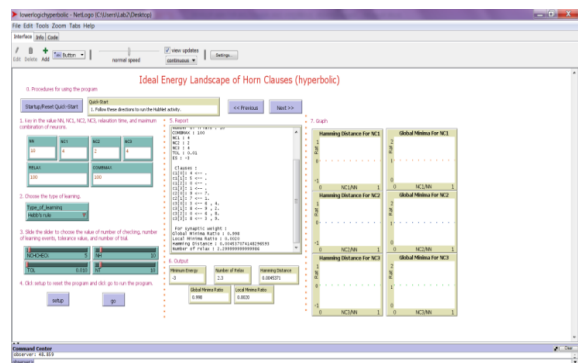


Figure 4: Agent Based Modelling for integration of hyperbolic activation function in logic programming of HONN

### VI. EXPERIMENTAL RESULTS AND DISCUSSION

We run the network by using two methods, Boltzmann machine and hyperbolic tangent activation function to accelerate the performance of doing logic programming. Thus a comparison of the Wan Abdullah method, Boltzmann machine and hyperbolic tangent activation function in logic programming in Hopfield networks is carried out.

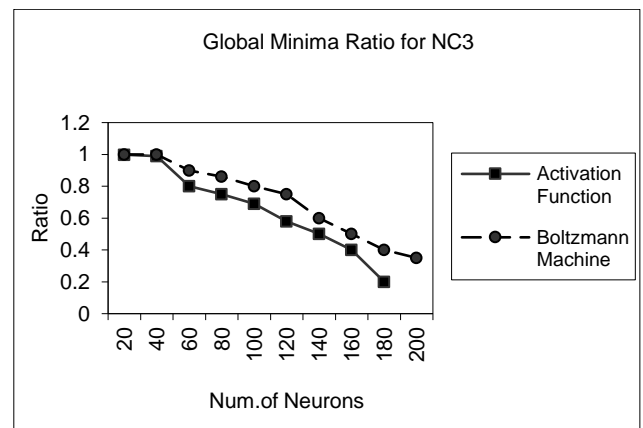


Figure 5: Global minima ratio for third order clauses

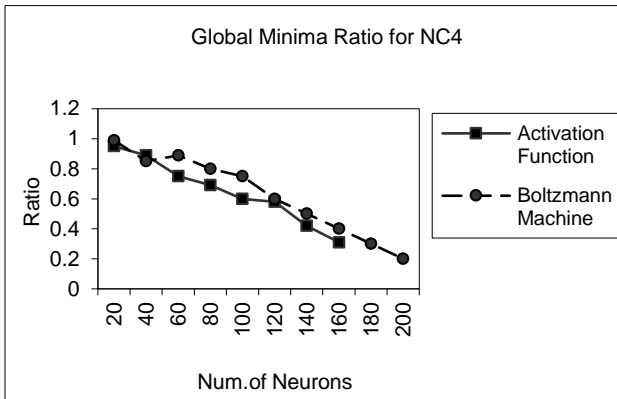


Figure 6: Global minima ratio for fourth order clauses

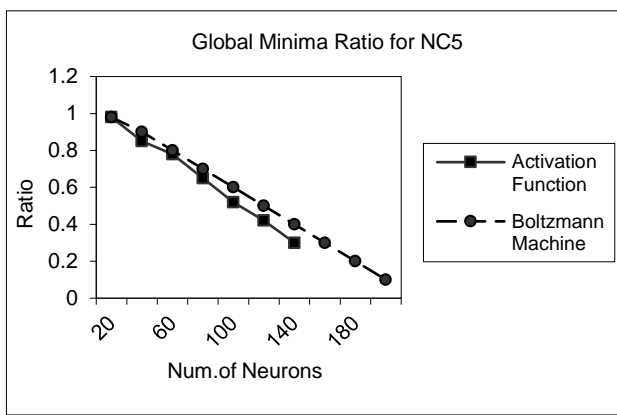


Figure 7: Global minima ratio for fifth order clauses

We simulated the network for the both methods. The following figures shows the global minima ratio (Number of global minima solutions/Number of runs) obtained by using both methods. From the graphs obtained, we observed that the ratio of global solutions is consistently 1 for all the cases, although we increased the network complexity by increasing the number of neurons (NN) and number of literals per clause (NC3, NC4, NC5) for the new activation function compare with the sigmoid function. It can be observed that when the network gets larger or more complex, the Boltzmann machine seems to perform better and continuously compared with the activation function. This is due to the simulated annealing procedure carried out by the Boltzmann machine where the neurons are forced to jump the energy barriers to relax into global solutions. By using this method the neurons are able to relax to global minima values rather than stuck in local minima values. While using activation function function,

the neurons get stuck and unable to jump the energy barrier to relax in global states.

Among the methods, Boltzmann method achieves the best value among. When the networks get larger or more complex, more neurons are applied in the program, Boltzmann method also have better performance than hyperbolic method and the Wan Abdullah method.. It is because there was an advantage when using Boltzmann machine, after running for long enough at a certain temperature, the probability of a global state of the network will depend only upon that global state's energy, according to a Boltzmann distribution. When temperature decreases from a high temperature to reach a thermal equilibrium at a low temperature, we are guaranteed to converge to a distribution where the energy level fluctuated around the global minimum. The heat cause the neurons unstuck from local minima. As a result, Boltzmann machine is having the best result among the others methods.

From the overall comparison in logic programming that perform higher order Horn clause, each method had achieved low and non-ideal global minima. However for Boltzmann machine, it produces better global minima among them.

## VII. CONCLUSION

From the study found that the ability of Boltzmann machine doing in logic program on Hopfield network is better than Wan Abdullah method (only McCulloch pits updating rule applied) and hyperbolic tangent activation function of logic program. It provides a better result in term of global minima ratio in generating higher horn clauses. Computer simulations that had been carried out verified the validity of the Boltzmann machine performance compare to new activation function. This completes our illustration of computer simulation to test the validity and strength of the proposed method of doing logic programming in Hopfield network.

### VIII. ACKNOWLEDGEMENT

This research is partly financed by Fundamental Research Grant Scheme (203/ PMATHS/6711368) by Ministry of Higher Education and Universiti Sains Malaysia.

### REFERENCES

- [1] J. J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proceedings National Academy of Science USA*, Vol. 79, 1982, pp. 2554-2558, 1982.
- [2] Pinkas, G., Energy minimization and the satisfiability of propositional calculus. *Neural Computation*, 3, pp 282-291, 1991.
- [3] Pinkas, G., Propositional no monotonic reasoning and inconsistency in symmetric neural networks. *Proceedings of the 12<sup>th</sup> International Joint Conference on Artificial Intelligence*, pp. 525-530, 1991.
- [4] Wan Abdullah, W.A.T., *Logic Programming on a Neural Network*. *Int .J. Intelligent Sys*, 7, pp. 513-519, 1992.
- [5] W. A. T. Wan Abdullah., Neural Network Logic. O. Benhar, C. Bosio, P. del Giudice and E.Tabet (eds.), *Neural Networks: From Biology to High Energy Physics*, ETS Editrice, Pisa, pp.135-142, 1991.
- [6] Brenton Cooper, Stability analysis of higher-order neural networks for combinatorial optimization, *International Journal of Neural Systems*, 12, pp. 177-186, 2002.
- [7] Ding, Y., Dong, L., Wang, L. and Wu, G., A High Order Neural Network to Solve Crossbar Switch Problem. In: Wong, K.W. et al. (eds.) *ICONIP 2010, Part II, LNCS 6444*, Heidelberg: Springer pp. 692–699, 2010.
- [8] Cheung, K.-W, Lee, T., Boundary Detection by Artificial Neural Network. *IEEE Proceeding of 1993 International Joint Conference on Neural Networks, Nagoya, 1993*. 2, pp. 1189-1194, 1993.
- [9] Kwok-wai Cheung, Tong Lee, *Boundary Detection by Artificial Neural Network*. International Joint Conference on Neural Networks, 1993.
- [10] Joya, G, Atencia, M.A. and Sandoval, F., Hopfield neural networks for optimization: study of the different dynamics, *Neurocomputing* 43, Amsterdam: Elsevier Science B.V pp.219-237, 2002.
- [11] Roweis S. (n.d.) *Boltzmann Machines* [Online]. [Accessed by 15 May 2013]. Available from: [www.cs.nyu.edu/~roweis/notes/botlz.pdf](http://www.cs.nyu.edu/~roweis/notes/botlz.pdf)
- [12] Sathasivam, S. and Wan Abdullah, W.A.T., The Satisfiability Aspect of Logic on Little Hopfield Network. *ISSN 1450-223X (7)*, pp.90-105, 2010.