Modern Applied Science; Vol. 8, No. 3; 2014
ISSN 1913-1844 E-ISSN 1913-1852
Published by Canadian Center of Science and Education

Boltzmann Machine and Hyperbolic Activation Function in Higher Order Network

Saratha Sathasivam[1] & Muraly Velavan[2]

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

[2] School of General and Foundation Studies, AIMST University, 08100 Bedong, Kedah, Malaysia

Correspondence: Saratha Sathasivam, School of Mathematical Sciences, Universiti Sains Malaysia Penang, 11800 USM, Malaysia. E-mail: saratha@usm.my

Received: February 9, 2014 Accepted: April 7, 2014 Online Published: April 23, 2014

doi:10.5539/mas.v8n3p140 URL: http://dx.doi.org/10.5539/mas.v8n3p140

Abstract

For higher-order programming, higher-order network architecture is necessary to provide faster convergence rate, greater storage capacity, stronger approximation property, and higher fault tolerance than lower-order neural networks. Thus, the higher-order clauses for logic programming in Hopfield Networks have been considered in this paper. The goal is to perform logic programming based on the energy minimization scheme is to achieve the best global minimum. However, there is no guarantee to find the best minimum in the network. However, by using Boltzmann Machines and hyperbolic tangent activation function this problem can be overcome.

Keywords: Higher Order Hopfield Network (HONN), Boltzmann machine and hyperbolic tangent activation function

1. Introduction

Neural Networks is a mathematical model or computational model that is inspired by the structure of biological neurons such as the brain process information. It can solve sophisticated recognition and analysis problems. It is because it composed of huge amount of interconnected neurons to solve specific problems. However in this paper, we are concentrated on Hopfield network. Hopfield network is a recurrent neural network (Hopfield, 1982) 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.

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 (Pinkas, 1991, 1992) and Wan Abdullah (Wan Abdullah, 1991, 1992). Further detail can refer to the references. The advantages by using Wan Abdullah's method are it can revolves around propositional Horn 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.

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. Meanwhile in section 4 contain discussions regarding the Boltzmann machine and Hyperbolic Tangent activation function. Finally, section 5 and 6 occupy the simulation results and concluding remarks regarding this work.

2. Higher Order Hopfield Networks

Discrete Hopfield network is shown in Figure 1, as 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.

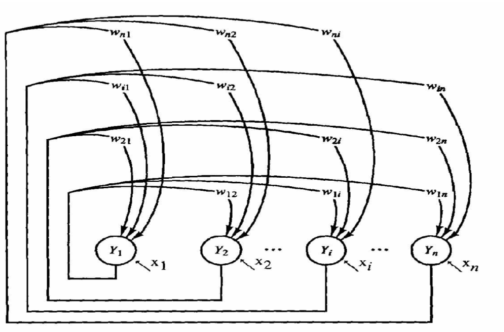

Figure 1. Discrete hopfield network

In neural network, higher order logic programming is highly regarded as the essential method in Hopfield Networks. When the Hopfield neural network is used to solve NP-complete optimization problem (Brenton, 2002; Ding et al., 2010; Cheung & Lee, 1993) such as travelling salesman problem, positive solutions would be produced. From literature review, few papers had carried out that applying higher order Hopfield networks such as using HOHN (Ding et al., 2010) to solve N-queens problem and construction method of energy function and neural computing method also shown. Besides, comparison with the first order Hopfield network and the method how to speed the convergence and escape from the local minima also had discussed in those papers. While according to Cheung and Lee (1993), the convergence property had restudied before put in application in real life. Besides, Ising spin problem also had carried out in it. The most important paper that affects the main backbone of this paper is explained by Joya et al. (2002). 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. However, in this paper only local minima and convergence are concentrated. Further explanation can refer to reference. From those stated above, Hopfield network has overcome the difficulty to find suitable parameters to guarantee convergence and explore a new path for artificial intelligence and intellectual computer.

Other than that, higher-order Hopfield network 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.

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

$$E = -\frac{1}{n}\sum_{i_1}\sum_{i_2}\sum_{i_3}\cdots\sum_{i_n} w_{i_1 i_2 i_3 \cdots i_n} x_{i_1}x_{i_2}x_{i_3}\cdots x_{i_n} - \frac{1}{n-1}\sum_{i_1}\sum_{i_2}\sum_{i_3}\cdots\sum_{i_{n-1}} w_{i_1 i_2 i_3 \cdots i_{n-1}} x_{i_1}x_{i_2}x_{i_3}\cdots x_{i_{n-1}} - \cdots - \frac{1}{2}\sum_{i_1}\sum_{i_2} w_{i_1 i_2} x_{i_1}x_{i_2} - \sum_i w_i x_i \qquad (1)$$

where x_i is the state value of i^{th} neuron , $w_i$defines the connection weights of the nth order connection from neurons $i_1 i_2 i_3 \cdots_{i_n}$ 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

$$x_i\,(t) = sgn(h_i(t)),$$

$$h_i = \sum_{i_1}\sum_{i_2}\sum_{i_3}\cdots\sum_{i_n} w_{i_1 i_2 i_3 \cdots i_n} x_{i_1}x_{i_2}x_{i_3}\cdots x_{i_n} + \sum_{i_1}\sum_{i_2}\sum_{i_3}\cdots\sum_{i_{n-1}} w_{i_1 i_2 i_3 \cdots i_{n-1}} x_{i_1}x_{i_2}x_{i_3}\cdots x_{i_{n-1}} + \cdots + \sum_{i_1}\sum_{i_2} w_{i_1 i_2} x_{i_1}x_{i_2} + \sum_i w_i x_i \qquad (2)$$

where sgn is signum function. The connection weight of higher-order Hopfield networks is symmetrical. This condition is analogical to the symmetric requirement of the Hopfield Netowrks connection weight matrix.

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 (Joya et al., 2010). Thus, the Boltzmann machine and Hyperbolic Tangent activation function will carry out in this paper.

3. Logic Programming

Logic programming is the use of mathematical logic for computer programming. Thus, higher order Hopfield network (HONN) 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← B, C as A ∨ ⌐ (B ∧ C) = A ∨ ⌐B ∨ ⌐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, $\frac{1}{2}(1-S_A)$; $E_p=\frac{1}{2}(1-S_A)\frac{1}{2}(1+S_B)\frac{1}{2}(1+S_C)+\ldots\ldots$ 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 $A<-B_1,B_2,B_3,\ldots\ldots,B_n$ 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.

4. Boltzmann Machine

Hopfield networks have recognized that some relaxation schemes have a joined cost function and the states of the network converge to local minima of this function. It had performed optimization of a well-defined function. However, there is no guarantee to find the best minimum in the network. Thus, Boltzmann Machines had introduced to overcome this problem. A Boltzmann machine is a network of units which are fully interconnected by bidirectional connections with symmetric weights. There are no self-connections are allowed. These units have binary values 0 and 1 that referring to states ON and OFF for each unit. The major difference from Hopfield networks is the way of updating the states which determined by stochastic decisions. Boltzmann machines have a simple learning algorithm that allows them to discover interesting features that represent complex regularities in the training data. Furthermore, it extends the concept of Hopfield networks by a stochastic update method.

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 (Sathasivam & Wan Abdullah, 2010). The difference in the global energy that results from a single unit *i* being 0 versus 1, written ΔE_i, is given by:

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

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:

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

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.

4.1 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.

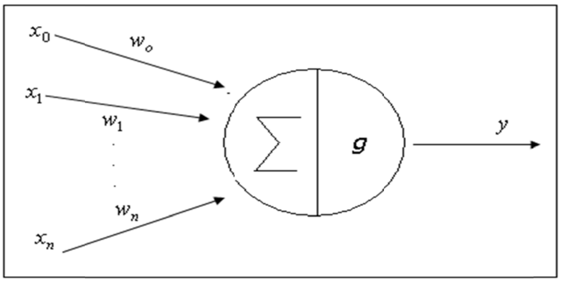

Figure 2. An artificial neuron with activation function

Figure 2 shows the artificial neuron x which is a neuron with n input dendrites $(x_0 \ldots x_n)$ and where $(w_0 \ldots w_n)$ 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.

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

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.

5. Experimental results and discussion

We used *C++* as a platform to simulate the programs.

5.1 Maximum Complexity

Maximum complexity means the maximum or the limitation of the system can be afford to run before oscillation occurred. In this paper, we want to figure out the maximum complexity that can be reached when higher order clause was applied in logic programming. We did trial and error technique to determine the maximum capacity for the system. From the technique we selected number of neurons (NN) = 30. The system started having overload error for running more than 30 neurons for higher orders. We used fifth dimension arrays to represent the higher order logic programming.

5.2 Hamming Distances

Hamming distance was originally conceived for detection and correction of errors. It is simply defined as the number of bits that are different between two bit vectors. It is easier to discuss closeness of a bit pattern to another bit pattern by determining the number of bit positions in which the two patterns being compared differs. Thus, at here hamming distance is measured between the final state and global state of the neurons upon relaxation.

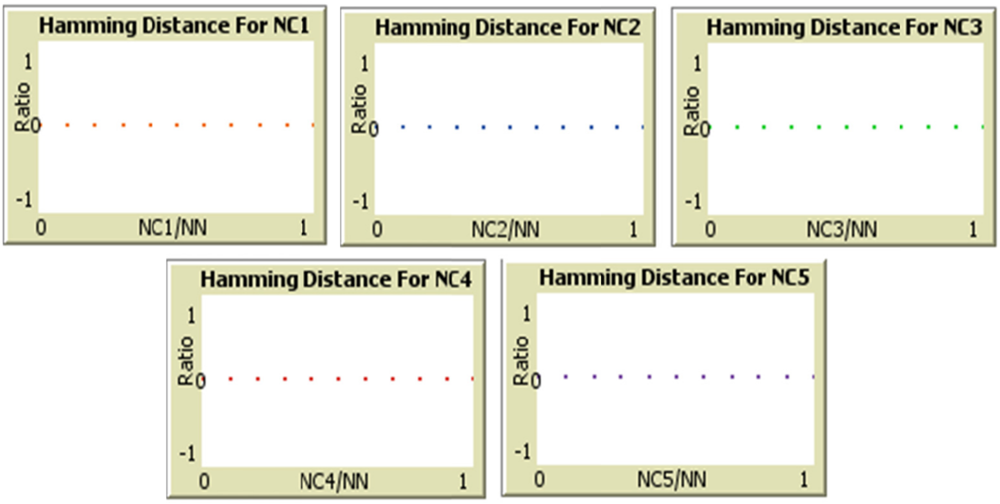

Figure 3. Hamming distance for NC1~NC5

In Figure 3, it shows that the hamming distance graphs of every level of clauses such as NC1, NC2, NC3, NC4 and NC5. It shows that the hamming distance is approximately to zero for all the cases, (NC1 (one literal), NC2 (two literals), NC3 (three literals), NC4 (four literals), NC5 (five literals)). The value of hamming distance was approximately to zero due to all the neurons reached a stable state. So, the distance between the stable states and global states are almost zero. Due to this, similar statistical error was obtained for all the cases.

5.3 Global Minima

From the global minima graphs, when the neurons relaxed to global solutions the ratio will approached 1.

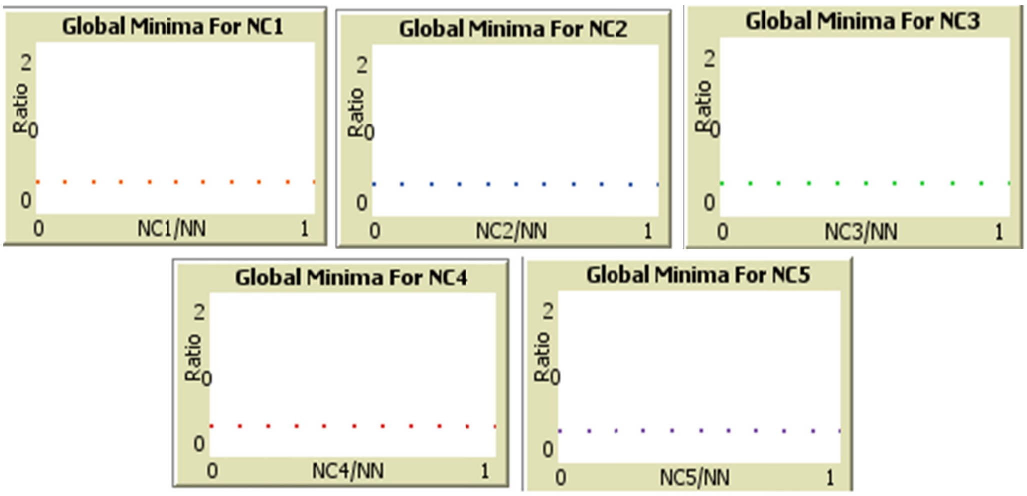

Figure 4. Graphs of global minima for NC1~ NC5

From the Figure 4 obtained, the ratio of global solutions is approximately to 0.375 for all cases when the number of neurons and number of literals per clause (NC1, NC2, NC3, NC4, and NC5) increased. It is because by using Higher Order Hopfield Network (HOHN), it was guaranteed to yield convergence towards toward local minima. When the network gets more complex due to increasing complexity, the neurons start to oscillate and stuck in local minima values.

Furthermore, the abrupt energy surface does not allow for the existence of local minima escape trajectories. This problem is increasing proportionally to the size of the error surface, and there is no universal solution (Joya et al., 2002). However, it stills have some ways to escape the local minima. One way to escape the local minima in order to find the global minimum is to resort to simulated annealing. Thus, Boltzmann machine and hyperbolic tangent activation function in higher order to show which method will enhance a better global minima.

5.4 Comparing Global Minima in Higher Order Logic Programming

We simulated 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.

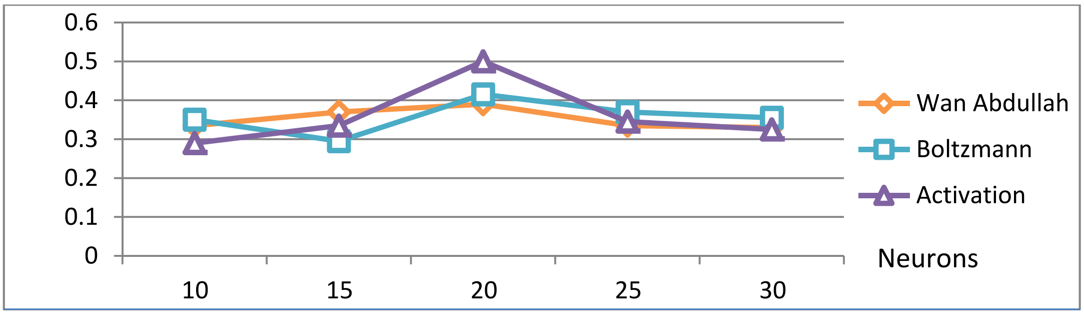

Figure 5. Global minima ratio show according to number of neurons in higher order logic programming

From the Figure 5, it shows that the global minimum ratios at first are different for each method. The ratio for Wan Abdullah method is 0.335, the Boltzmann method is 0.35 and hyperbolic method is 0.29. All the global minima values obtained are not more than 0.4. This problem occurs for higher order Hopfield network. To address this problem, the application of arbitrary order Hopfield neural networks to optimization problems is studied (Joya et al., 2002). This paper studied about the main problems affecting practical applications of higher order Hopfield networks such as existence of local minima and others. From the paper, it had made justification that a high probability of evolution to local minima occurs when involving higher order Hopfield networks. One of the simulation results for the evolution of a higher order Hopfield networks oriented for solving a Diophantine equation had shown that the percentage of correct solution (to achieve global minima) is only 38% (almost the same ratio value when run for fifth order clause). To help to get through the problem of local minima, Boltzmann machine and hyperbolic tangent activation functions are implemented to enhance the global minima. 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 when using Boltzmann machine, after running for a long enough time 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.

6. Conclusion

From the theory and experimental result, the ability of Boltzmann machine in doing logic programming on Hopfield network is better than Wan Abdullah method, which is based on Mc Culloch Pitts updating rule, and hyperbolic tangent activation function. It provides a better result in term of global minima ratio and hamming distance for higher order clauses. However, for higher order, the result of global minima that obtained was very low due to the complexity of network.

Acknowledgement

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

References

Brenton, C. (2002). Stability analysis of higher-order neural networks for combinatorial optimization. *International Journal of Neural Systems, 12*, 177-186. http://dx.doi.org/10.1142/S0129065702001151

Cheung, K. W., & Lee, T. (1993). Boundary Detection by Artificial Neural Network. *IEEE Proceeding of 1993 International Joint Conference on Neural Networks, Nagoya, 2*, 1189-1194.

Ding, Y., Dong, L., Wang, L., & Wu, G. (2010). A high order neural network to solve crossbar switch problem. In *Neural Information Processing, Models and Applications* (pp. 692-699). Springer Berlin Heidelberg.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings National Academy of Science USA, 79*, 2554-2558. http://dx.doi.org/10.1073/pnas.79.8.2554

Joya, G., Atencia, M. A., & Sandoval, F. (2002). Hopfield neural networks for optimization: study of the different dynamics. *Neurocomputing, 43*, 219-237. http://dx.doi.org/10.1016/S0925-2312(01)00337-X

Pinkas, G. (1991a). Energy minimization and the satisfiability of propositional calculus. *Neural Computation, 3*, 282-291. http://dx.doi.org/10.1162/neco.1991.3.2.282

Pinkas, G. (1991b). Propositional no monotonic reasoning and inconsistency in symmetric neural networks. *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (pp. 525-530).

Roweis, S. (n.d.). *Boltzmann Machines*. Retrieved May 15, 2013, from www.cs.nyu.edu/~roweis/notes/botlz.pdf

Sathasivam, S., & Wan Abdullah, W. A. T. (2010). The Satisfiabilty Aspect of Logic on Little Hopfield Network. *American Journal of Scientific Research, 7*, 90-105.

Wan Abdullah, W. A. T. (1991). Neural Network Logic. In O. Benhar, C. Bosio, P. del Giudice & E. Tabet (Eds.), *Neural Networks: From Biology to High Energy Physics* (pp. 135-142). Pisa: ETS Editrice.

Wan Abdullah, W. A. T. (1992). Logic Programming on a Neural Network. *Int. J. Intelligent Sys, 7*, 513-519. http://dx.doi.org/10.1002/int.4550070604

Copyrights