

# Reverse Analysis in Higher Order Hopfield Network for Higher Order Horn Clauses

Saratha Sathasivam, Ng Pei Fen and \*Muraly Velavan

School of Mathematical Sciences  
Universiti Sains Malaysia, Penang, Malaysia

\*School of General and Foundation Studies  
AIMST University, Bedong, Kedah, Malaysia

Copyright © 2014 Saratha Sathasivam, Ng Pei Fen and Muraly Velavan. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## Abstract

In Reverse Analysis, we can learn the inherent relationships among the data by extract common patterns that exist in dataset. We can look for unknown and unexpected relations, which can be uncovered by exploring of data sets. Moreover, Reverse Analysis can use as data mining for tool. In this paper, an agent based modelling (ABM) was introduced by using Netlogo which carry out higher order horn clauses representation of Reverse Analysis in doing Hopfield network. Our interest in this paper is confined largely to an important class of neural networks that perform useful computations through a process of learning. So, from the ABM that designed, we can carry out some computer simulation to verify and test the ABM develop.

**Keywords:** Higher Order Hopfield Network, Netlogo, agent based modeling

## 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 [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 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 Reverse Analysis, it is a method that when examining the connection strengths obtained during the learning process, logical rules that have been acquired may be deduced. This step will repeat until it gets the value that wanted. From the Reverse Analysis given the values of the connections of a network, we hope to know what logical rules are entrenched in it.

In this paper, we develop agent based modelling (ABM) in Higher Order Hopfield Networks for generate higher order Horn Clauses by applying Reverse Analysis. 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. Meanwhile in section 4 contain discussions regarding the Reverse Analysis. In section 5, the introduction about Netlogo and agent based modeling will carry out. Finally, section 6 and 7 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.

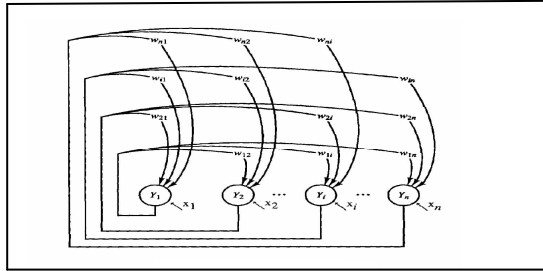


Figure 1: Discrete Hopfield network

In neural network, higher order logic programming is highly regarded as the essential method in Hopfield Networks which is used to solve NP-complete optimization problem [6, 7, 8, and 9] such as travelling salesman problem, positive solutions would be produced. Hopfield network has overcome the difficulty to find suitable parameters to guarantee convergence and explore a new path for artificial intelligence and intellectual computer. Furthermore, higher-order Hopfield networks can be used to 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.

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 \$i^{th}\$ neuron.

\$w\_{i\_1 i\_2 i\_3 \dots i\_n}\$ defines the connection weights of the nth 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

$$x_i(t) = \text{sgn}(h_i(t)), \tag{2}$$

$$h_i = \sum_{i_1} \sum_{i_2} \sum_{i_3} \dots \sum_{i_n} w_{i_1 i_2 i_3 \dots i_n} x_{i_1} x_{i_2} x_{i_3} \dots x_{i_n} + \sum_{i_1} \sum_{i_2} \sum_{i_3} \dots \sum_{i_{n-1}} w_{i_1 i_2 i_3 \dots i_{n-1}} x_{i_1} x_{i_2} x_{i_3} \dots x_{i_{n-1}} + \dots + \sum_{i_1} \sum_{i_2} w_{i_1 i_2} x_{i_1} x_{i_2} + \sum_i w_i x_i$$

where  $\text{sgn}$  is signum function. The connection weight of higher-order Hopfield networks is symmetrical.

### 3. Logic Programming

Logic programming is the use of mathematical logic for computer programming. In this paper, higher order Hopfield network (first order connections to fifth order connections) had been carried out in logic programming model. A HOHN is used to minimise logical inconsistency in interpretations of logic programs and clauses. logic program play in the system. Following is the logic program that built by using Wan Abdullah's method in Hopfield network. Following is the algorithm:

- 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. In the next section, an introduction of Reverse Analysis will carry out.

### 5. Reverse Analysis in higher order Hopfield networks

Users would not limit their jobs in the simple network that involve lower connections among neurons. It is not enough for more information if only third order connections are involved in the system. Thus, in this section, higher order Reverse Analysis will introduce to run for higher

connections. Theory is the same with the lower order connections [10] but in this section will have fifth order connections among the neurons.

The following flow diagram shows how Reverse Analysis has been applied in higher order Hopfield network.

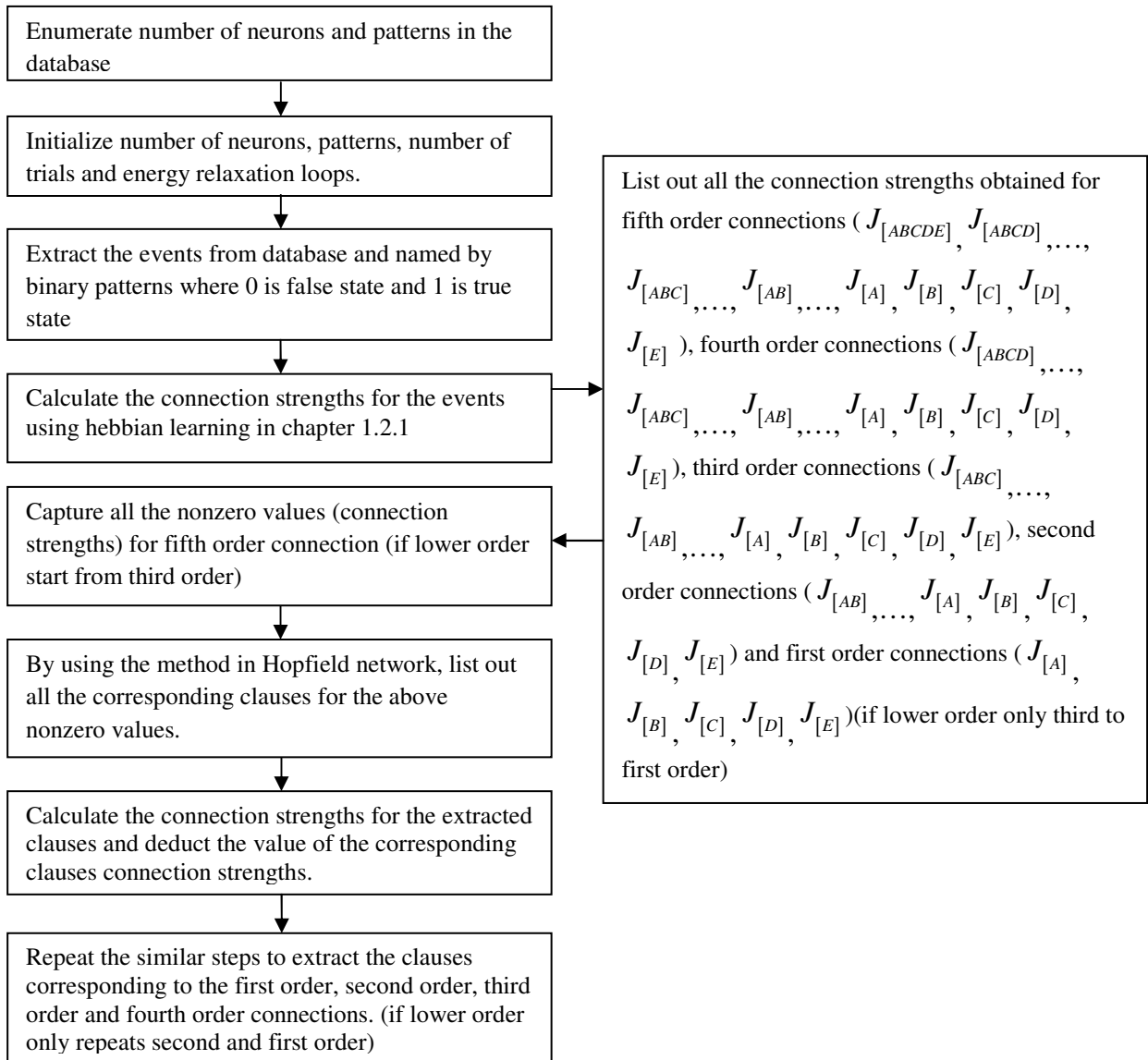


Table 1: Logical clauses connections strengths using Wan Abdullah's method in higher order clauses

Logical clauses	Connection strengths								
	$J_{[ABCDE]}$	$J_{[ABCD]}$	$J_{[ABC]}$	$J_{[AB]}$	$J_{[A]}$	$J_{[B]}$	$J_{[C]}$	$J_{[D]}$	$J_{[E]}$
A←B, C, D, E	$\frac{1}{64}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{32}$	$-\frac{1}{32}$	$-\frac{1}{32}$	$-\frac{1}{32}$	$-\frac{1}{32}$
A←B, C, D	-	$\frac{1}{32}$	-	$\frac{1}{16}$	$\frac{1}{16}$	$-\frac{1}{16}$	$-\frac{1}{16}$	$-\frac{1}{16}$	-
A←B, C	-	-	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{8}$	-	$-\frac{1}{8}$	-	$-\frac{1}{8}$
A←B	-	-	-	$\frac{1}{4}$	$\frac{1}{4}$	-	$-\frac{1}{4}$	-	-
A←	-	-	-	-	$\frac{1}{2}$	-	-	-	-
←B	-	-	-	-	-	$-\frac{1}{2}$	-	-	-
←C	-	-	-	-	-	-	$-\frac{1}{2}$	-	-
←D	-	-	-	-	-	-	-	$-\frac{1}{2}$	-
←E	-	-	-	-	-	-	-	-	$-\frac{1}{2}$

## 6. Introduction to Netlogo

After learnt the efficient of Reverse Analysis, we will use Netlogo [11, 12] to create a simulator for the user. In this simulator, programmer needs to design an agent based modeling to implement the higher order Hopfield network in Reverse Analysis. Netlogo is a multi-agent programming language and integrated modeling environment. There has been continuous development for connected learning and computer-based modeling. Furthermore, it is a well suited method for modeling complex systems that can give instructions to hundreds or thousands of “agents” to operate independently like turtle. It is because it is fully programmable and formed by simple language structure. It can instruct mobile agents move over a grid of stationary agents. 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. The model can be viewed in either 2D or 3D form.

Programmer can choose any interesting agent shapes to design the agent based modeling, and some interface builders like buttons, sliders, switches, choosers, monitors, notes, output area in the agent based modelling can be developed too. Those interface builders are ready to use and programmer do not need to write more programming language from them. Later in the next section will carry out the definition and more explanation of simulator and benefits of agent based modeling in Netlogo.

## **6.1 Model of Neural Networks Simulator**

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. 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.

Moreover, agent-based Modelling (ABM) [13, 14] which also called individual-based modelling 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. 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. It is because, the agent based modelling paradigm are commonly used in dynamics and complex communities such as telecommunication, health care and others that involving large populations which used explicitly capture social networks.

In general, when we build an ABM to simulate a certain phenomenon, we need to identify the actors first (the agents). We then need to consider the processes (rules) governing the interactions among the agents. Figure 2 shows the ABM that was built for this purpose.

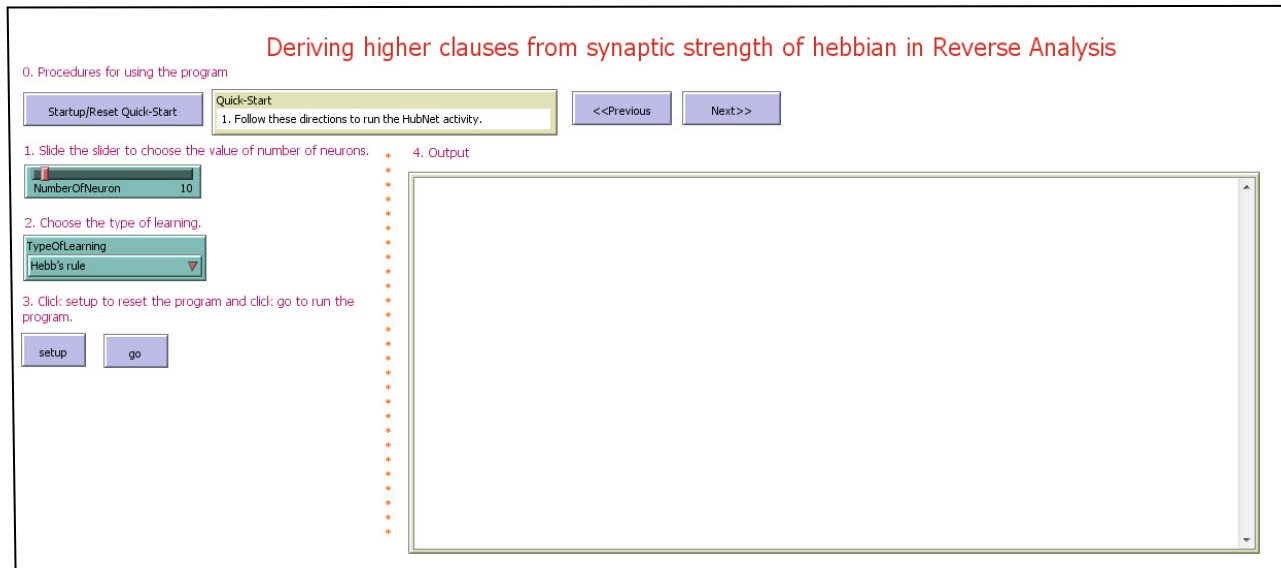


Figure 2: The layout of deriving higher clauses from synaptic strength of hebbian in Reverse Analysis

Explanation of the flow chart:

#### PHASE 1: Entering Values

- 1) Press the start up / Reset Quick-Start button for the new user.
  - a. User can press the next button to go to the next step and previous button to go to the previous step.
- 2) Later, key in Number of neurons.
  - a. The maximum of NN is 30 in higher order. These values are set after trial and error method had applied.
- 3) Choose type of learning which is either Hebb's rule or Wan Abdullah's method [15, 16].
- 4) After all the value had been set, press the setup button to fix and set those values in the program.
- 5) Then, press go button to run the program.
  - a. User input message will pop out to ask the user to store events in binary number (0, 1). It depends on how many numbers of neurons you declared. For example if number of neurons is 10, the user input message will pop out for 10 times asking the user key in binary number. This ABM is letting the user to declare the value of clauses.
  - b. After that, user choice message will pop out to ask the user decide whether want to continue to store another set of event or not. If yes will pop up user input message again, if no will start to run the program. From here, a set of events are stored to have a pattern for the system to predict the clauses.



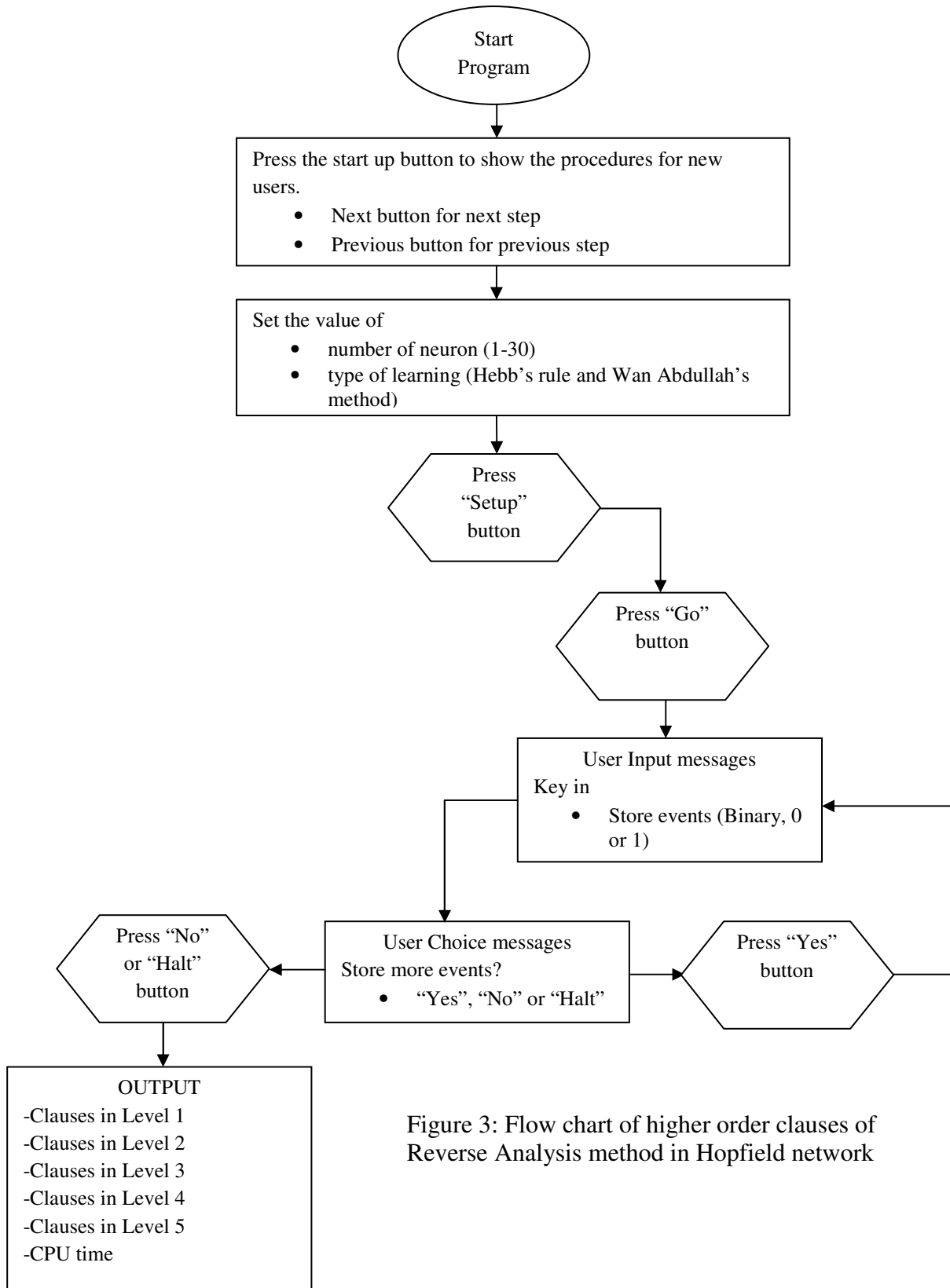


Figure 3: Flow chart of higher order clauses of Reverse Analysis method in Hopfield network

## PHASE 2: Training

- 6) Enumerate number of neurons and pattern in the system and initialize the value that had key in by user in the clauses
  - a. Next, the system will extract out the events that users had entered and named them according to 0 and 1. From here, the connection strengths for the events will be calculated using Hebbian learning. After that, system will list out all the connection strengths obtained from fifth order connections to first order connections. Later by capture the nonzero in level 5 then sorting the neurons by declare them in fifth dimension array and based on the condition that declared, if condition fulfil, the clause will assume pass (neurons formed will declare as a clause) and undergo deduct value process. However, if fail, if will go to the next level to check the condition had fulfilled or not. If until level 1 it still fails to fulfil, the clause will not generate. It will start again by running another way of neurons arrangement. After all the clauses in every level had formed, the testing is coming to its end.

Lastly, the system will print out the output of each run.

## 7. Experimental results and discussion

For deriving higher order clauses in Reverse Analysis, the time needed for generating the clauses depend on the value theuser enter for each event. Moreover, from the clauses that have been generated, we can also able to predict the relationships between the neurons. However for generation of clauses higher than the fifth order, the system seems to oscillate. Due to this, we limit our analysis up to fifth order only.

### I. Maximum Complexity

The program can support up to 30 neurons after applying try and error technique. For more than 30 neurons, the memory overloading and high computational complexity problem arise.

### II. Centre Processing Unit (CPU) time

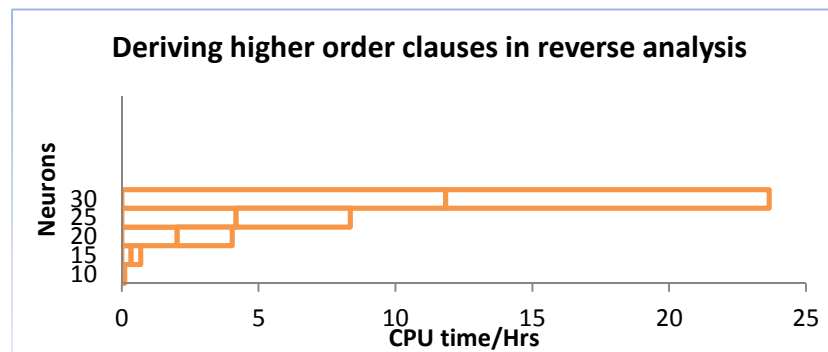


Figure 5: CPU time taken according to the number of neurons in higher classes

From figure 5, it had shown the CPU time was getting larger when network gets larger and more complex. The complexity of the network increased as the network gets bigger.

## **8. Conclusion**

In this paper, we had developed agent based modeling to carry out Reverse Analysis in Hopfield network by using NETLOGO as a platform which is user friendly. By using ABM, the user can view and analyse the Reverse Analysis method in 3-D effectively. Although ABMs develop for the Reverse Analysis method in Hopfield network are quite efficient, the system still facing oscillation problem when the network get,

## **ACKNOWLEDGEMENT**

This research is partly financed by Science Fund grant (305/ PMATHS/613142) from the Ministry of Higher Education, Malaysia and USM's short term grant (304/PMATHS/6312053).

## **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.
- [2] G. Pinkas, Energy minimization and the satisfiability of propositional calculus. *Neural Computation*, 3, 1991, pp 282-291.
- [3] G. Pinkas, Propositional nonmonotonic reasoning and inconsistency in symmetric neural networks. *Proceedings of the 12<sup>th</sup> International Joint Conference on Artificial Intelligence*, 1991, pp 525-530.
- [4] W.A.T.Wan Abdullah, *Logic Programming on a Neural Network*. Int .J. Intelligent Sys, 7, 1992, pp 513-519.
- [5] W.A.T.Wan Abdullah, Neural Network logic. *Proc of the workshop held in Elba International Physics Center, Italy*, 1991, pp 135-142.
- [6] T. Maxwell, C. L. Giles, Y. C. Lee, and H. H. Chen, Transformation Invariance Using High Order Correlations in Neural Net Architectures, *IEEE Trans. Syst. Man Cybern. SMC*, 1986, CH2364-8, 627.

- [7] H. H. Chen, Y. C. Lee, T. Maxwell, G. Z. Sun, H. Y. Lee, and C. L. Giles, High Order Correlation Model for Associative Memory, *AIP Conf. Proc.* 151, 1986, 86.
- [8] Y. C. Lee, G. Doolen, H. H. Chen, G. Z. Sun, T. Maxwell, H. Y. Lee, and C. L. Giles, Machine Learning Using a Higher Order Correlation Network, *Physica D*, 1986, 22, 276.
- [9] P.H. Wen, A Review of Hopfield Neural Networks for Solving Mathematical Programming Problems, *European Journal of Operational research*, Vol.198, Num.3, 2008, 675-688.
- [10] S. Sathasivam, Upgrading Logic Programming in Hopfield Network, *Sains Malaysiana*, Vol.39, Num.1, 2010, 115-118.
- [11] <http://ccl.northwestern.edu/netlogo/docs/>. Accessed on 03/08/2013.
- [12] <http://en.wikipedia.org/wiki/NetLogo>. Accessed on 13/09/2013.
- [13] Doi: 10.1073/pnas.082080899 *PNAS May 14, 2002 vol. 99 no. Suppl 3 7280-7287*
- [14] [http://en.wikipedia.org/wiki/Agent-based\\_model](http://en.wikipedia.org/wiki/Agent-based_model) . Accessed on 13/08/2013.
- [15] D.O. Hebb, The organization of Behaviour. 1949, *New York: John Wiley*.
- [16] S. Sathasivam. *Logic Mining in Neural Network*. PhD Thesis, 2007, Malaysia.

**Received: October 15, 2013**