Journal of Genetics and Genetic Engineering
Volume 2, Issue 3, 2018, PP 7-14
ISSN 2637-5370

Activation Function of Hopfield Neural Network in Agent Based Modeling

[1]ShehabAbdulhabib Saeed Alzaeemi , [2] AdilZamil Abed , [3] SarathaSathasivam

[1]*School of Mathematical Sciences, UniversitiSains Malaysia, 11800 USM, Penang Malaysia*
[2]*Directorate of Anbar education, Ministry of education of Iraq*
[3]*School of Mathematical Sciences, UniversitiSains Malaysia, 11800 USM, Penang Malaysia*

***Corresponding Author:** ShehabAbdulhabib Saeed Alzaeemi ,School of Mathematical Sciences, UniversitiSains Malaysia, 11800 USM, Penang Malaysia Email: shehab_alzaeemi@yahoo.com*

ABSTRACT

In recent years, most of the researchers in the field of artificial intelligence focused on developing the Agent Based Modelling in order to simplified their learning algorithms. This version of simplification is useful to find a provable criterion for convergence in a dynamic system. In this paper, an agent based modelling (ABM) was developed by using NETLOGO as a platform for activation functions to carry out logic programming in Hopfield network. The developed model seems to illustrate the task of doing logic programming in a simple, flexible and user friendly manner.

Keywords: *Hopfield network, NETLOGO, Agent Based Modelling , Logic Program*

INTRODUCTION

There are numerous neural networks which provide different performance for different applications. Common neural network architectures include Radial Basis network, Single layer network, Multilayer network, Competitive network, and Hopfield network. Hopfield network is invented by John Hopfield in 1982 that consist of a set of N interconnected neurons is connected to each other in both directions. It consists of a synaptic connection (synaptic weight) pattern that is the building blocks of energy function (Lyapunov function) for dynamic activities [1, 2]. In our mind, to apply logic program will be embedded in the Hopfield network. Logic program provides a natural way for solving the problems [3]. Logic programming can be demarcated as an optimization problem for horn clauses or 2-SAT or k-SAT [13, 14] that is similar to other optimization problem such as traveling salesman problem. Logic program with the agent beside modeling is easier to understand, easier to verify and also easier to change [7]. Logic programming in Hop field network was presented by Wan Abdullah and revolve around propositional Horn clauses [12, 15]. GadiPinkas and Wan Abdullah [11, 12] proposed a bi-directional mapping between logic and energy function in a symmetric neural network. The first agent based modelling to carry out logic programming in Hopfield network by using NETLOGO as platform was proposed by Sathasivam [8].

In this paper, we develop agent based modeling (ABM) for activation functions to do logic programming in Hopfield network. Agent-based modelling (ABM) alternatively called individual-based modelling or multi-agent based modelling. It is a powerful simulation modelling techniques that have gained recognition in a number of applications in the last few years, including applications to real-world business problems. In agent-based modelling (ABM), a system is modelled as a collection of autonomous decision-making agents. Agent-based model (ABM) is kind of micro scale model that simulate the simultaneous operations and interactions of multiple agents in an attempt to re-create and predict the appearance of complex phenomena. The process is one of emergence from the lower (micro) level of systems to a higher (macro) level. As such, a key notion is that simple behavioral rules generate complex behavior. Another central tenet is that the whole is greater than the sum of the parts. Individual agents are typically characterized as boundedly rational, presumed to be acting in what they perceive as their own interests, such as reproduction, economic benefit, or social status, using heuristics or simple decision-making rules [16]. ABM agents may experience "learning", adaptation, and reproduction. As these non-linear, adaptive interactions are mostly too complex to be captures by analytical expression, computer simulations are most often use, idea of such simulation is to specify the rules of behavior of individual entitles, as well as the rules of their interaction in a multitude of the individual entitles using a computer model and to explore the consequences of the specified individual-level rules on the level of population as a whole, using results of simulation of their behavior and interaction are known as agent-based stimulation, the properties of individual agents describing their behavior and interaction are known as elementary properties, and the properties emerging on the higher, collective level are known as emerging properties.

Agent- based modelling is appealing and interesting because of the collective level are often neither obvious nor expectable, even in many cases when the assumption on individual agent properties are very simple, the capability of generation complex and intriguing emergent properties arises not so much that the in-built rules of individual agent behavior, as from the complexity of the network of interaction among the agents. Precisely this multitude of agents, as well as the multitude and complexity of their interaction are the main reasons why in most cases formal mathematical deduction of results of an agent based model is not possible ABMs are also called individual-based models (IBMs), and individuals within IBMs may be simpler than fully autonomous agents within ABMs. [17]. A review of recent literature on individual-based models, agent-based models, and multiagent systems shows that ABMs are also useful in non-computing related scientific domains including biology, ecology and social science. Agent-based modelling is related to, but distinct from, the concept of multi-agent systems or multi-agent simulation. ABM is also use to search for explanatory insight into the collective behavior of agents obeying simple rules, typically in natural systems, rather than in designing agents or solving specific practical or engineering problems. It combines elements of game theory, complex systems, emergence, computational sociology, multi-agent systems, and evolutionary programming. Monte Carlo methods are used to introduce randomness.

The rest of the paper is organized as follows. Section II Contains Hopfield Neural Network. In Section III, We briefly discussed Logic Programming In Hopfield Network. IV. we talk about Activation Function Section V we discuss the significance and the usage of NETLOGO. Section VI describes Agent Based Modelling (ABM) in Hopfield network. Section VII discuss the implementation of Agent Based Modelling in logic programming. Section VIII discussion and section IX to conclude the study.

Hopfield Neural Network

Hopfield neural network is easier to be integrated with any paradigms to solve satisfiability problem [19, 20, 7]. A Hopfield network is a network of N interconnected artificial neurons, which are fully interconnected [1,21]. The connection weight from neuron j to neuron i is represented and denoted by the number w_{ij} , in general, the number w_{ij} is symmetric, that is $w_{ij} = w_{ji}$ and in Hopfield network $w_{ii} = w_{jj} = 0$ (no self connection), the set of all such numbers is represented by the connection weight matrix W, whose elements are w_{ij}. The local field of the connection is given by

$$h_i = \sum_j W_{ij}^{(2)} S_j + J_i^{(1)} \tag{1}$$

Where … denotes the higher order connection. The updating rule reads

$$S_i(t+1) = \mathrm{sgn}[h_i(t)] \tag{2}$$

In order to check the correctness of the final state, total energy of the neurons will be evaluated. An energy function for the Hopfield network is given by [6]:

$$E = -\frac{1}{2}\sum_i \sum_j W_{ij}^{(2)} S_i S_j - \sum_i W_i^{(1)} S_j \tag{3}$$

The Hopfield network has demonstrated the interesting features [5,13,14,22]:

- Distributed representation: A store memory.
- Distributed asynchronous control: All of these local actions add up to a global solution.
- Content addressable memory: The network will automatically find the closest match.
- Fault tolerance: The network can still function properly.

Logic Program (By Wan Abdullah Method) Hopfield Neural Network

A logic program consists of program clauses and is activated by an initial goal statement [3]. Moreover, logic programming is a high-level language, easy to create a prototype, and offer shorter and more readable programs that suit many AI applications. Logic programming is a programming paradigm based on formal logic. A program written in a logic programming language is a set of sentences in logical form, expressing facts and rules about some problem domain [23]. It is made up a set of axioms and a goal statement. The program allows user to state a collection of axioms from which a theorem or goal can be proven [24]. When the goal is stated by the user, the language implementation will search a collection of axioms and inference steps that together imply the goal. In almost all logic languages, axioms are written in a standard form known as Horn clause. A Horn clause consists of a head and a body:

$$H \leftarrow B_1, B_2, B_3, \ldots B_n \tag{4}$$

Where H is the head and B_i are the body. Horn clause is a clause of disjunction of literals that have at most one positive literal. The below expression is a typical example of Horn clause:

$$\neg A \vee \neg B \vee \ldots \vee \neg E \vee H \tag{5}$$

Where the positive literal in the expression is H, whereas the others literals are negative since the symbol (¬) is negation of the literal that follows it. A Horn clause that consists of exactly one positive literal is called a definite Horn clause.

Equation (5) above can be rewritten equivalently in a simplified form as shown below.

$$(A \wedge B \wedge \ldots \wedge E) \rightarrow H \tag{6}$$

The lexicon (non-logical symbol) of propositional logic consists of a set of proposition symbols. In equation (5) above, the proposition symbols are A, B, ..., E and H. The logical symbols of propositional logic are: ¬, ↔, and →. Following are some terminologies of propositional logic in logic programming.

A sentence of the form $A \vee B$ is a disjunction.

A sentence of the form $A \wedge B$ is a conjunction

A statement of the form $A \rightarrow B$ is an implication, where A is refers to as antecedent and B is termed consequent.

A statement of the form¬A means negation of A (Not A) this will change the state of A from whatever state it is earlier

Based on Wan Abdullah's method, the following algorithm summarizes how a logic program can be done in a Hopfield network proposal by Wan Abdullah [9,32]:

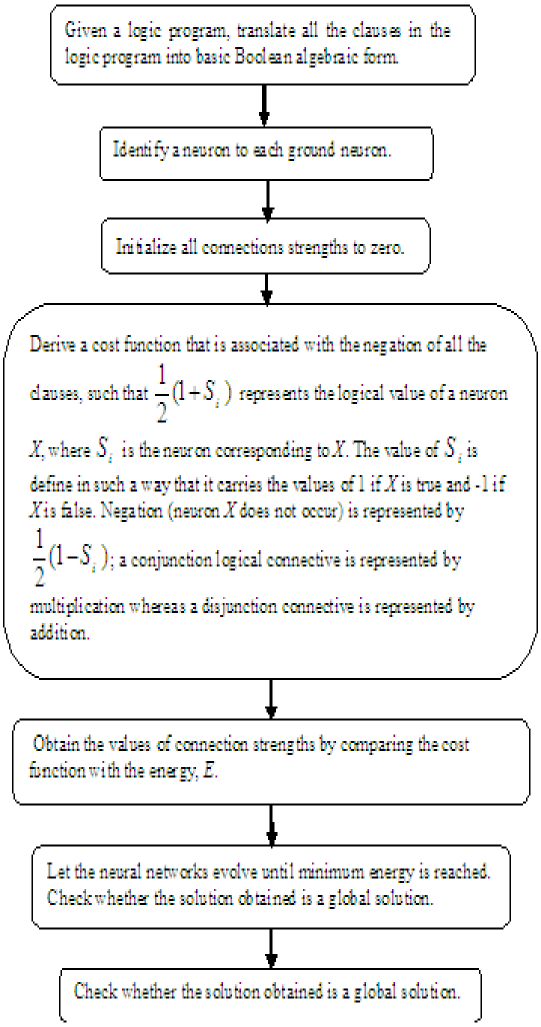

Figure1. *Flow chart on algorithm how a logicProgram can be done in a Hopfield network.*

Activation Function

The first activation function implemented in logic programming in Hopfield neural network was the sign function by McCulloch-Pitts (ideal model) proposed by Walter and Pitts [34]. Although McCulloch-Pitts Activation Function helps the network to find global solution, this function is prone to few weaknesses such as computational burdening and lack of efficiency in producing desired results. McCulloch-Pitts neuron has been generalized in many ways and one of the obvious generalizations is to use activation functions other than threshold function. Figure 2 below shows different types of activation functions:

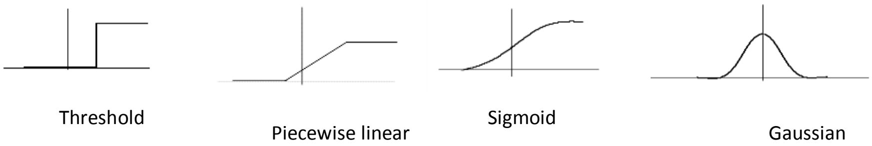

Figure2. *Various Types of Activation Function*

In computational networks, the activation function defines the output of the neuron by the given input. The activation function for the original McCulloch-Pitts neuron was the unit step function. However, the artificial neuron model has since been expanded to include other functions such as the sigmoid, piecewise linear and transfer function. The activation function is sometimes called a “transfer”. The most commonly used transfer functions are logistic sigmoid function and hyperbolic tangent function. According to Kalveram [33], the logistic sigmoid function was being more frequently used than the hyperbolic tangent function in Hopfield neural network. Numerous researchers proved Hyperbolic activation function outperform most of the activation function [35, 36].

NetLogo

To support the development of agent-based models (ABM), a number of different platforms have been developed [4]. These platforms differ in how many assists they provide. Most of the factors in the program of the NetLogo are represented by Mason and Swarm. Agent besides Modeling offers a set of software libraries to be used in programming a model. NetLogo programming consists of a programming language (derived from the previously LOGO language) and a set of libraries, as well as a programming environment. However, NetLogo supplies a graphical tool for quickly constructing interfaces for running agent-based models. When using NetLogo there are a lot of advantages. One of the advantages is its interface. More information on the interface can be found in the NetLogo interface guide [24, 25]. NetLogo is also an authoring environment. NetLogo is simple enough to enable students and researchers to create their own models, even if they are not professional programmers [26]. NetLogo has a simple interface, models can be set up and run with only the push of one or two buttons; to produce the model such as the one illustrated takes just 50 lines of simple code. NetLogo is good for setting up simple simulations very quickly [27, 28, 29, 30, 31]. In this paper, we focus on NetLogo as a powerful research tool that is also suitable for learners at the undergraduate level and above.

The Agent Based Modelling

We designed agent-based modeling for the implementation of the activation function in the Hopfield Network for doing logic programming by using Wan Abdullah methods. Using NetLogo as a platform for developing agent-based modeling (ABM) . NetLogo was designed and authored by Uri Wilensky[10]. Increasing used Agent-Based Modeling (ABM) in a broad range of social sciences. Agent-Based Modeling involves building a computational model consisting of agents, each of which represents an actor in the social world, and an environment [18].

Implementation Of Agent Based Modelling In Logic Program

Sathasivam [8] developed the first reverse analysis in Hopfield network. With that in mind, Agent Based Modelling (ABM) doing logic programming in Hopfield network will be presented.

Agent-Based Modeling (ABM) designed to implement the activation functions in the Hopfield network for doing logic programming. Thus, a computer program which emulates exactly what the user want needs to construct in order to simulate the action of Hopfield Network. An agent-based modeling had designed for the user to run the simulator. Moreover, agent-based Modelling which also called individual-based modeling is a new computational modeling paradigm which is an analyzing system that representing the agents that involving and simulating of their interactions. In this section the steps involved in developing ABM for the logic programming method will be considered. Looking into the steps involved in developing the Agent Beside Modeling for activation functions to do logic programming in Hopfield Network as Figure

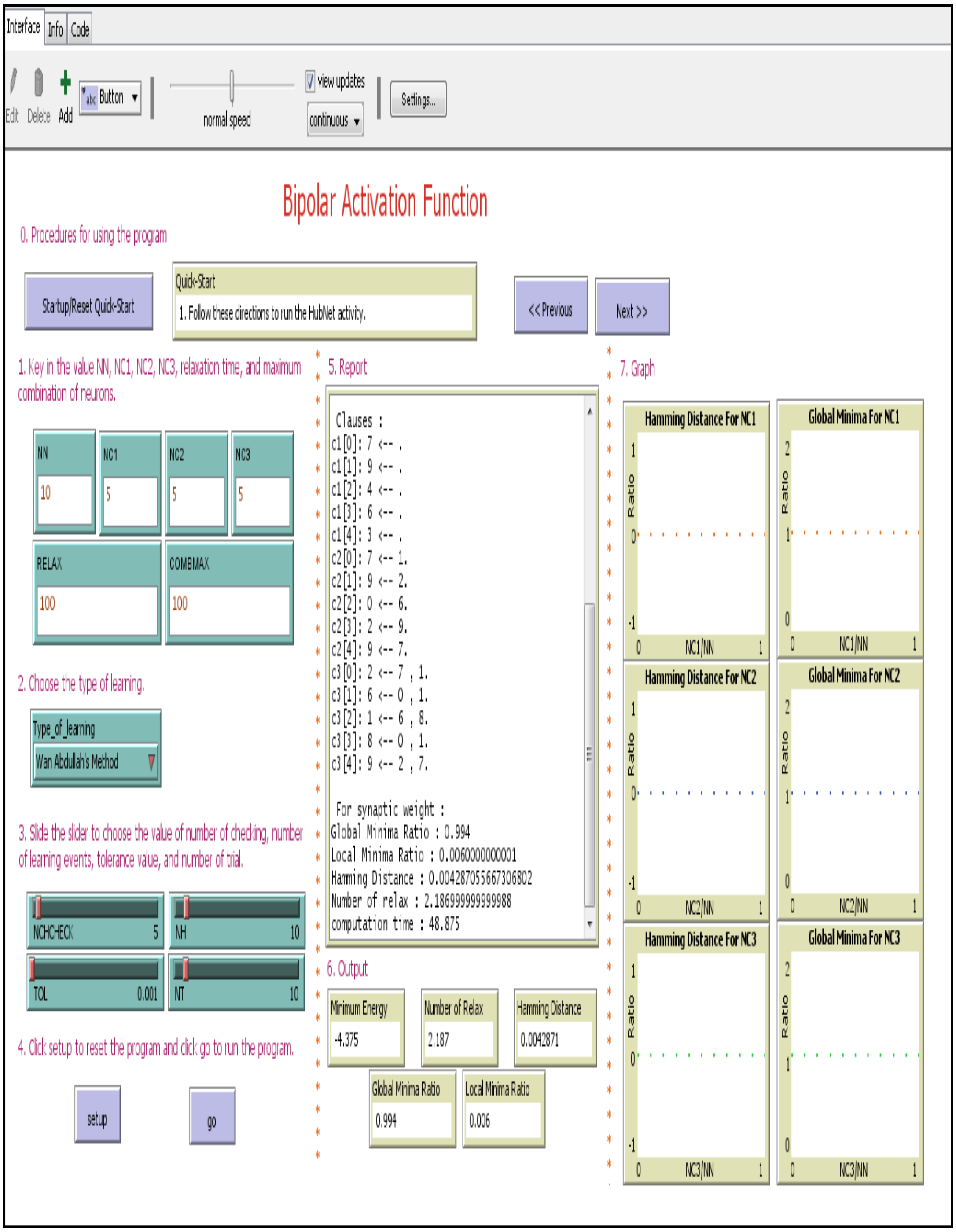

Figure3. *Layout of Agent Based Modelling using Netlogo*

DISCUSSION

We designed the agent-based modeling and training it with computer simulation. We generate a set of 3 random clauses with 4 variables and subject 10 neurons network through Wan Abdullah's Method from events satisfying the generated clauses. The development of agent-based modeling was done using windows 7, 64-bit, with the following specification (4096MB RAM, processor 3.40GHz, and 500GB hard disk) since the computer specification play a significant role in the performance of the Agent-Based Modelling (ABM). The developed ABM was designing in such a way that latest version of NetLogo (5.3.1), tools and techniques were utilized. The interface of the ABM was designed in such a way the programmer/ user see the series of procedures and stages involves so he/she has the flexibility to adjust input parameters at the beginning of the program and see the results of the hamming distance, global minima, computational time to mention but few at the end of the simulation in figure 3. By using agent-based modeling for activation functions to do logic programming in Hopfield Network. This shows that we can implement activation functions to do logic programming in Hopfield Network model through a specific procedure. Furthermore, using ABM, user can analyze the graphical design of the links more efficiently and systematically.

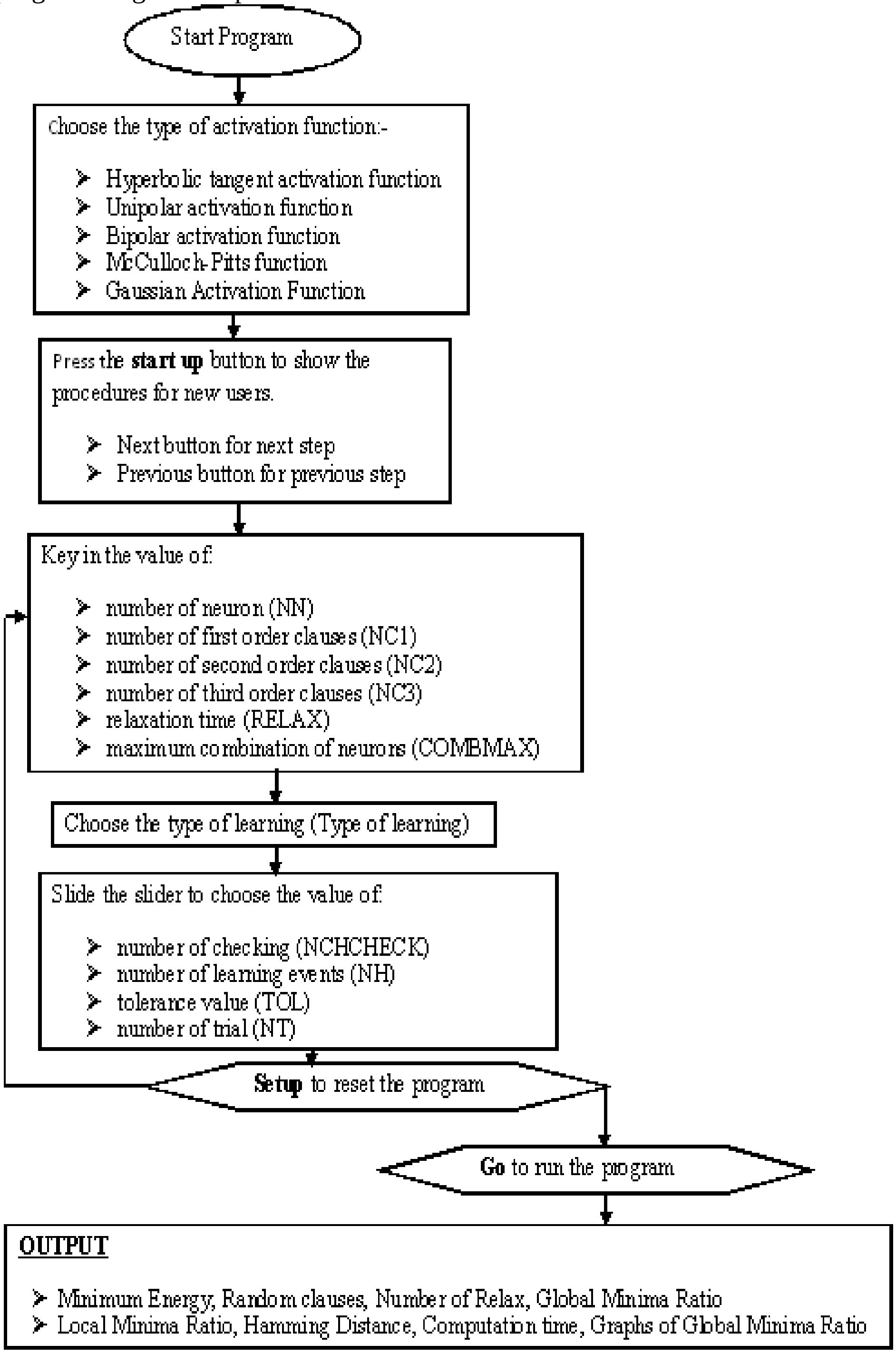

Figure4. *Flow Chart of Agent Based Modelling for Doing Logic Programming in Hopfield Neural Network*

CONCLUSION

In this paper, we had developed agent based modeling for activation functions to do logic programming in Hopfield Network by using NETLOGO as platform. Agent Based Modelling (ABM)is easy to handle but designed in different ways that give the users know more in Netlogo such as the user input message, read data systems that ease the users to keys in and 2D animation that had carried out. They were very user friendly. Besides, the benefits of ABM can be summarized as below:

ABM is able to integrate or link logic program and Hopfield Network. It offers a natural description of a system.

ABM is able to produce models for the set of logic program because it captures emergent phenomena.

ABM is flexible. Users can change the training parameters.

Although, ABM develops model, the process for activation functions to do logic programming in Hopfield Network are quite efficient and the system still faces oscillation problem when involving in high complexity of system. Thus, to improve it a future work will carry out.

References

[1] J.J. Hopfield. Neural Networks and Physical System with Emergent Collective Computational abilities. Proc.Natl. Acad. Sci. USA, 79,pp 2554-2558, 1982.

[2] J.J. Hopfield. Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons.Proceeding. Natl. Acad. Sci. USA. 81(10), pp 3088-3092, 1985.

[3] R.A. Kowalski. Logic Programming for Problem Solving, New York: Elesvier Science Publishing Co, (1979) .

[4] M. J. Berryman. Review of software platforms for agent based models. Technical Report DSTO-GD-0532, Defence Science and Technology Organisation, Edinburgh, Australia (April, 2008).

[5] S. Sathasivam. Logic Mining in Neural Networks. PhD. Thesis. University of Malaya, Malaysia , (2006).

[6] S. Sathasivam, & W.A.T. Wan Abdullah. Logic Learning in the Hopfield Networks, Modern Applied Science, 2(2), pp 57-62, 2008.

[7] S. Sathasivam. (2009), Energy Relaxation for Hopfield Network With the New Learning Rule. International Conference on Power and Optimazation, Bali.

[8] S. Sathasivam, & N. P. Fen. Developing agent based modeling for doing logic programming in hopfield network. Applied Mathematical Sciences,7(1), pp 23-35, 2013.

[9] W. A. T. Wan Abdullah . Logic Programming in Neural Networks. Malaysian Journal of Computer Science, 9(1), pp. 1-5, 1996.

[10] U. Wilensky, W. Rand, and Kornhauser, D "Visualization tools for agent-based modeling in NetLogo." Agent2007, Chicago, November , pp 15-17, (2007).

[11] W.A.T. Wan Abdullah, Logic Programming on a Neural Network. Malaysian Journal of computer Science, 9 (1), pp. 1-5, 1993.

[12] G. Pinkas, R. Dechter, Improving energy connectionist energy minimization, Journal of Artificial Intelligence Research, 3, pp. 223-15, 1995.

[13] S. Sathasivam, Upgrading Logic Programming in Hopfield Network, Sains Malaysiana, 39, pp. 115-118, 2010.

[14] S. Sathasivam, Learning in the Recurrent Hopfield Network, Proceedings of the Fifth International Conference on Computer Graphics, Imaging and Visualisation, pp. 323-328, 2008.

[15] R. Rojas, Neural Networks: A Systematic Introduction. Berlin: Springer, 1996.

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

[17] MACAL, Charles M.; NORTH, Michael J. Tutorial on agent-based modeling and simulation. In: Proceedings of the 37th conference on Winter simulation. Winter Simulation Conference, pp. 2-15, 2005.

[18] G.N. Gilbert. Agent-based models (No. 153). Sage, 2008 .

[19] D. Vilhelm, J. Peter, & W. Magnus, Counting models for 2SAT and 3SAT formulae. Theoretical Computer Science, 332 (1), pp. 265-291, 2005.

[20] Mansor, Mohd Asyraf, Mohd Shareduwan M. Kasihmuddin, and Saratha Sathasivam. "VLSI Circuit Configuration Using Satisfiability Logic in Hopfield Network." International Journal of Intelligent Systems & Applications 8.9 (2016).

[21] J.J. Hopfield. Neurons with Graded Response Have Collective Computational Properties like Those of Two-State Neurons. Proceeding. Natl. Acad. Sci. USA. , 81(10), pp 3088-3092, 1984.

[22] AUGUST, Mayer; GERALD, W.; MARKUS, S. Applications of Hopfield Networks. University of Salzburg-Institute for Computer Science, Austria, 1999.

[23] L.S. Michael, Programming Language Pragmatics. United States: Morgan Kauffman Publications, 2008.

[24] U. Wilensky, NetLogo Simulation Software northwestern. edu/netlogo Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL, 2008.

[25] A. M. Ramanath and N. Gilbert, The design of participatory agent-based social simulations, Journal of Artificial Societies and Social Simulation. 7(4) ,October, 2004.

[26] Tisue, Seth, and Uri Wilensky. "Netlogo: A simple environment for modeling complexity." International conference on complex systems. pp 16-21. 2004.

[27] U.Wilensky, Paradox, programming, and learning probability: A case study in a connected mathematics framework. The Journal of Mathematical Behavior, 14(2), pp. 253-280, 1995.

[28] M. Resnick, Beyond the centralized mindset. The journal of the learning sciences, 5(1), pp. 1-22, 1995.

[29] U. Wilensky,., & M. Resnick, Thinking in levels: A dynamic systems approach to making sense of the world. Journal of Science Education and technology, 8(1), pp. 3-19, 1999.

[30] U. Wilensky, Statistical mechanics for secondary school: The GasLab multi-agent modeling toolkit. International Journal of Computers for Mathematical Learning, 8(1), pp. 1-41, 2003.

[31] A.Ioannidou, A. Repenning, , C. Lewis, , G. Cherry, , & C. Rader, Making constructionism work in the classroom. International Journal of Computers for Mathematical Learning, 8(1), pp. 63-108, 2003.

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

[33] K. H. Kalveram, A Neural Network enabling sensorimotor learning. Neurocomputing. 55(1), pp. 299-314, 1992.

[34] W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics. 1, 5(4), pp. 115-33, 1943.

[35] M. S. Kasihmuddin, and S. Sathasivam, Accelerating activation function in higher order logic programming. In ADVANCES IN INDUSTRIAL AND APPLIED MATHEMATICS: Proceedings of 23rd Malaysian National Symposium of Mathematical Sciences (SKSM23) AIP Publishing.1750(1), p. 030006, 2016.

[36] M. A. Mansor, S. Sathasivam, Performance analysis of activation function in higher order logic programming. In ADVANCES IN INDUSTRIAL AND APPLIED MATHEMATICS:Proceedings of 23rd Malaysian National Symposium of Mathematical Sciences (SKSM23) AIP Publishing.1750(1), p. 030007, 2016.

Citation: *ShehabAbdulhabib Saeed Alzaeemi , AdilZamil Abed , SarathaSathasivam." Activation Function of Hopfield Neural Network in Agent Based Modeling ", Journal of Genetics and Genetic Engineering, vol. 2, no.3, pp. 7-14 2018.*