ACTIVATION FUNCTIONS IN NEURO SYMBOLIC INTEGRATION USING AGENT BASED MODELLING

Saratha Sathasivam

School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia, saratha@usm.my

Muraly Velavan

School of General and Foundation Studies, AIMST University, Bedong, Kedah, Malaysia,dsmuraly@yahoo.com

---

Abstract: Logic program and neural networks are two important aspects in artificial intelligence. This paper is part of an endeavour towards neural networks and logic programming integration. The goal in performing logic programming based on the energy minimization scheme is to achieve the best ratio of global minimum. However, there is no guarantee to find the best minimum in the network. To achieve this, activations functions are modified to accelerate the neuro symbolic integration. These activation functions will reduced the complexity of doing logic programming in Hopfield Neural Network (HNN).The activations functions discussed in this paper are new learning rule, Mc Culloch Pitts function and Hyperbolic Tangent Activation function. This paper also focused on agent based modelling for presenting performance of doing logic programming in Hopfield network using various activation functions. The effects of the activation function are analyzed mathematically and compared with the existing method. Computer simulations are carried out by using NETLOGO to validate the effectiveness on the new activation function. The resuls obtained showed that the Hyperbolic Tangent Activation function outperform other activation functions in doing logic programming in Hopfield network. The models developed by agent based modelling also support this theory.

Keywords: Neuro-symbolic, Logic programming, Hopfield, Activation function, Agent based modelling

---

Introduction: Neural network is a parallel processing network which generated with simulating the image intuitive thinking of human, on the basis of the research of biological neural network, according to the features of biological neurons (such as our brain) and neural network and by simplifying, summarizing and refining [1-3]. It uses the idea of non-linear mapping, the method of parallel processing and the structure of the neural network itself to express the associated knowledge of input and output.

The Hopfield neural network [4] is a simple recurrent network which can work as an efficient associative memory, and it can store certain memories in a manner rather similar to the brain. Wan Abdullah [5] proposed a method of doing logic program on a Hopfield network where optimization of logical inconsistency is carried out by the network after the connection strengths are defined from the logic program; the network relaxes to neural states which are models (i.e. viable logical interpretations) for the corresponding logic program. Type of learning implemented in this network is known as Wan Abdullah's learning or Direct Learning Rule. The connection weights are determined by comparing the cost function with energy function of the network. 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.

Logic Programming On A Hopfield Network: In this section we will discussed briefly some important concepts related in doing logic programming in Hopfield network by using program clauses. Take into consideration a system consists of *N* formal neurons and each of represented by an Ising variable $S_i(t), (i = 1, 2, ...N)$. Neurons then are bipolar, $S_i \in \{-1, 1\}$, follow the dynamics $S_i \to \mathrm{sgn}(h_i)$, where the field, $h_i = \sum_j J_{ij}^{(2)} S_j + J_i^{(1)}$, *i* and *j* running over all neurons *N*, $J_{ij}^{(2)}$ is the connection strength from neuron *j* to neuron *i,* and $-J_i$ is a fixed bias (negative of the threshold) applied externally to neuron i. Hence, the neuron modifies its state S_i; according to Mc-Culloch Updating Rule.

The connections are symmetric and zero-diagonal, $J_{ij}^{(2)} = J_{ji}^{(2)}, J_{ii}^{(2)} = 0$, which allows one to write a Lyapunov or energy function,

$$E = -\frac{1}{2} \sum_i \sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \tag{1}$$

which monotonically decreases with the dynamics. The dynamics thus allows the handling of combinatorial optimization problems, whereby neurons are mapped onto the combinatorial variables and the cost function is equated to the energy function of the optimization problem.

The two-connection model can be generalized to include higher order connections. This modifies the "local field" into

$$h_i = ... + \sum_j \sum_k J_{ijk}^{(3)} S_j S_k + \sum_j J_{ij}^{(2)} S_j + J_i^{(1)} \tag{2}$$

where "....." denotes higher orders connections, and an energy function can be written as follows:

$$E = ...-\frac{1}{3}\sum_i\sum_j\sum_k J_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i\sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \quad (3)$$

provided that $J_{ijk}^{(3)} = J_{[ijk]}^{(3)}$ for *i, j, k* distinct, with [...] denoting permutations in cyclic order, and $J_{ijk}^{(3)} = 0$ for any *i, j, k* equal. The updating rule maintains

$$S_i(t+1) = \mathrm{sgn}[h_i(t)] \quad (4)$$

In logic programming, a set of Horn clauses which are logic clauses of the form $A \leftarrow B_1, B_2, .., B_n$ where the arrow is read as "if" and the commas "and", is given and the aim is to find 'models' corresponding to the given logic program. The model here refers to a setoff interpretation which satisfies the logical clauses.

Logic programming is the use of mathematical logic for computer programming. Thus, higher order Hopfield network had carried out in logic programming model [6, 7].

New Learning Rule (NLR): The activation function in the Hopfield network is the sigmoid function (equation 4). However this activation function puts too much emphasis on minor noise perturbation instead of the signals related to the cost and the constraints encoded in the network. Zeng and Martinez[8] proposed a new activation function as followed:

$$V_{X_i} = \frac{0.5(1+\tanh(\frac{U_{X_i}+x_o}{u_o}))}{1+\tanh(\frac{x_0}{u_0})} \quad (U_{X_i} < 0)$$
$$V_{X_i} = \frac{\tanh(\frac{x_0}{u_0}) + 0.5(1+\tanh(\frac{U_{X_i}-x_o}{u_o}))}{1+\tanh(\frac{x_0}{u_0})} \quad (U_{X_i} \geq 0) \quad (5)$$

where the parameters are defined as followed: V_{X_i} = activation function, U_{X_i} =initial states, x_o represents the threshold for V_{X_i} to become steep, and u_0 measures the steepness of the activation function. This function can tolerate with noise and do perform well when the network gets larger.

Hyperbolic Tangent Activation Function(HTAF): Hyperbolic Activation Function is a commonly used activation function in ANN. The Hyperbolic Activation Function has a derivative which can be used with gradient descent based training methods. This activation function will act as an efficient squashing function. According to Sunderajoo [9], Hyperbolic Activation Function is being utilized in many ANN because the function can avoid the network from collapse into a simple linear function. This activation function produce well defined output (between -1 and 1) [14]. Basically, the input of the neurons is updated according to the following equation (4).

where $h_i(t)$ is the local field of the network. Hyperbolic function which is $m(h)$ is given as followed

$$m(h_i) = \frac{e^{h_i} - e^{-h_i}}{e^{h_i} + e^{-h_i}} \quad (6)$$

The updating rules for the state of the neurons are incorporated with Hyperbolic Activation Function which is given as followed:

$$S_i(t+1) = \begin{cases} 1 \ \ for \ \ m(h_i) \geq 0 \\ -1 \ \ for \ \ m(h_i) < 0 \end{cases} \tag{7}$$

Agent Based Modelling: Firstly, a simulator of Hopfield networks that using a conventional computer had been created with a new network design or store a new set of memories. We used NETLOGO version 6.0 as the platform. 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) which also called individual-based modelling is a new computational modelling paradigm in Hopfield network. Their attributes and behaviours will be grouped 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 easier to understand and to be user friendly. In addition, ABM reveals the appearance of the systems from low to high level outcomes. Thus, 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 procedures of doing logic programming in Hopfield network.

Results And Discussion: We will discuss the performance of the models developed by using simulation computation time (CPU). From the Figure 1 it can be observed that overall for Lower Order Network the three methods also perform quite good in relaxing into solutions. This is because the network complexities still low and the neurons able to relax smoothly to global solutions. Since hyperbolic function is used to update the neurons state in HTAF so the probability for the neurons to get stuck in local minima becomes less. The neurons flipping and fluctuation can be minimized by using this activation function.

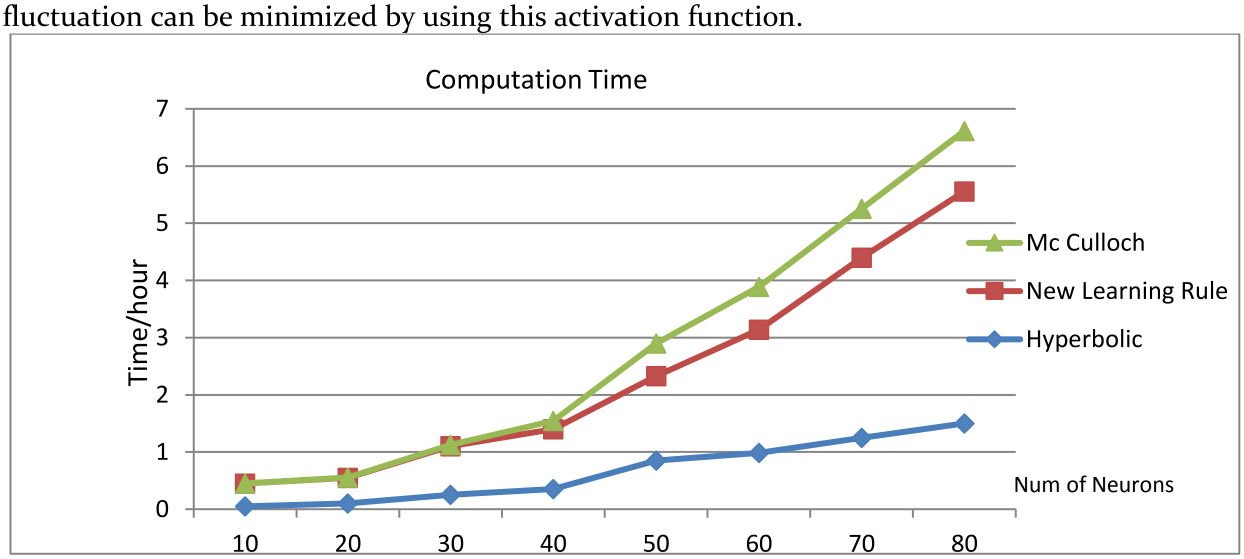

Figure 1: CPU time for training

HTAF seems to perform better that NRL and McCulloch. This is due to the usage of hyperbolic function. NRL performs better than McCulloch Pitts since its using parameter modifications.

Conclusion: In this paper, we had developed agent based modeling to carry out logic programming in Hopfield network by using NETLOGO as a platform. From the study we found that the ability of MFT in doing logic program on Hopfield network is good. It provides a better result in term of hamming distances. Although ABMs develop model the process of doing logic programming in Hopfield network are quite efficient, the system still facing oscillation problem when involving in high complexity of systems that larger than third order clauses.

Acknowledgment: The research is financed by USM GOT grant (1001/PMATHS/823032) and Universiti Sains Malaysia.

References:

[1] Rajan, R., Ankita, S. A Comparative Study on Variational Iteration Method and Modified Variational Iteration Method, Mathematical Science International Research Journal, 3(1), 2014, pp 347-350.

[2] Bunthu, K., Arundhati, K. & Pathan, J. Comparative Analysis of Machine Learning Techniques for Software Realibity Prediction, Mathematical Science International Research Journal , 3,2014, pp 977-992.

[3] DiPankar, DE. Fusion of Fuzzy Systems and Genetic Algorithms, Mathematical Science International Research Journal, 5(1), 2016, pp 108-112.

[4] Hopfield , J.J. Neural networks and physical systems with emergent collective computational abilities, *Proceedings National Academy of Science USA*, Vol. 79, 1982, pp. 2554-2558.

[5] Wan Abdullah, W.A.T. *Logic Programming on a Neural Network*. Int .J. Intelligent Sys, 7, 1992, pp 513-519.

[6] Kasihmuddin, B. M., Shareduwan, M., Mansor, B., Asyraf, M., & Sathasivam. Hybrid Genetic Algorithm in the Hopfield Network for Logic Satisfiabiality Problem. *Journal of Science and Technology, Pertanika*, 25(1), 2016, pp 139-152.

[7] Kasihmuddin, B. M., Shareduwan, M., Mansor, B., Asyraf, M., & Sathasivam, S. Genetic Algorithm for Restricted Maximum k-Satisfiability in the Hopfield Network. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4(2), 2016, pp 52-60.

[8] Zeng, X. & Martinez, R. A new activation function in the Hopfield Network for Solving Optimization Problems. Proceedings of Fourth International Conference On Artificial Neural Networks and Genetic Algorithms, Slovenia, 1999, pp 73-77.

[9] Sunderajoo. T, “New Algorithm in Training Neural Networks”, Msc.Thesis, Universiti Sains Malaysia, 2013.