16th International Conference on Researches in Science & Technology
(ICRST), 14-15 July 2017, Bali, Indonesia

AGENT BASED MODELLING FOR NEW TECHNIQUE IN NEURO SYMBOLIC INTEGRATION

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

This paper shows on developing agent based modelling for represent the performance of doing logic programming in Hopfield network by using a new activation function. The effects of the activation function on the performance of the neuro-symbolic integration are analyzed mathematically and compared with the existing method. Computer simulations are carried out to validate the effectiveness on the new activation function. The resuls obtained showed that the new activation function outperform the existing method 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

1. 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 and neural network and by simplifying, summarizing and refining. 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 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 (Wan Abdullah, 1992) proposed a method of doing logic program on a Hopfield network.

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. The connection weights are determined by comparing the cost function with energy function of the network.

In this paper, we will be using developed agent based modeling to analyze the usage of new activation function in enhancing the performance of doing logic programming in Hopfield network.

2. Hopfield network

The Hopfield network (Hopfield, 1982) is a single layer recurrent network that embodies the idea to storing information as the stable states of a dynamically evolving network configuration. Using an energy function in terms of the connection weights and output of the neurons, Hopfield [showed how much such networks can be used to solve specific problem in associative memory and combinatorial optimization. The discrete Hopfield network (DHNN) is used as associative memory, in which stored data is recalled by association with input data, rather than by an address.

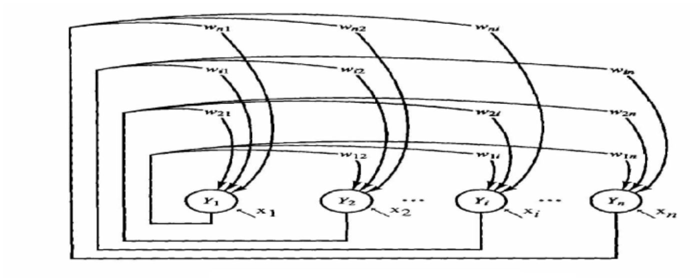

Figure 1: Discrete Hopfield Model

Fig. 1 shows a Hopfield network of N neurons. The input vector is $X = [x_1, x2, .x_n]$, and the state of the network is given by the output vector $Y = [y_1, y2, .y_n]$, where y_i denotes the output of neuron i, which can only be ± 1. Hopfield network is uniquely defined by (**W**, **b**), there into, $W = [w_{ij}]$ is a N x N-dimensional zero –diagonal matrix, and $[w_{ij}]$ is the weight connecting neuron i and j. B is a N –dimensional vector, where b_i denotes the fixed threshold value of each neuron i. and the relation of them is shown as followings:

$$\begin{cases} x_j(t) = \sum_{j=1}^{N} w_{ij} y_j(s) + B_j \\ y_j(t+1) = \mathrm{sgn}(x_j(t)) \end{cases} i, j = 1, 2, ...N \qquad (1)$$

In (1), the output of neuron i am given by *sgn (.)*, which is a symmetric signum function, whose output is +1 or -1. if the argument of the signum function is zero, then the output of neuron i remains unchanged.

As an associative network, the operation of Hopfield network has two phases: the storage phase and the retrieval phase. The evolvement of Hopfield network belong s to a complex, nonlinear dynamic system. When simple asynchronous updating is used for the Hopfield network outputs, an energy function E (Lyapunov function) is given by (2):

$$E = -\frac{1}{2} \sum_{\substack{i=1 \\ i \neq j}}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} W_{ij} x_i(s) x_j(s) - \sum_{i=1}^{N} B_i y_i(s) \qquad (2)$$

As the network evolves according to the dynamics in (2), the energy E can only decrease or stay unchanged at each update. This is because the change ΔE duo to a change in the output y_i can only be zero or negative. Eventually the network will converge to a (local) minimum energy state because E is bounded from below. The local minimum points in the energy landscape correspond to the prototype patterns stored in the storage phase. It means Hopfield network retrieval approximately equal to the target patterns. In the next section, we will be looking at Hebbian learning for Hopfield networks. Hebbian learning is been used to calculate the synaptic strengths between the neurons.

3. Logic Programming In Hopfield Network

In order to keep this paper self-contained we briefly review the Little-Hopfield model . The Hopfield model is a standard model for associative memory. The Little dynamics is asynchronous, with each neuron updating their state deterministically. The system consists of N formal neurons, each of which is described by an Ising variable. Neurons then are bipolar, obeying the dynamics $S_i \to \mathrm{sgn}(h_i)$, where the field, $h_i = \sum_j J_{ij}^{(2)} V_j + J_i^{(1)}$, i and j running over all neurons N, $J_{ij}^{(2)}$ is the synaptic strength from neuron j to neuron i, and $-J_i$ is the threshold of neuron i.

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

$$E = -\frac{1}{2}\sum_i \sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \qquad (3)$$

which monotone decreases with the dynamics.

The two-connection model can be generalized to include higher order connections. This modifies the “field” to be

$$h_i = ... + \sum_j \sum_k J_{ijk}^{(3)} S_j S_k + \sum_j J_{ij}^{(2)} S_j + J_i^{(1)} \qquad (4)$$

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

$$E = ... - \frac{1}{3}\sum_i \sum_j \sum_k J_{ijk}^{(3)} S_i S_j S_k - \frac{1}{2}\sum_i \sum_j J_{ij}^{(2)} S_i S_j - \sum_i J_i^{(1)} S_i \qquad (5)$$

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, and that similar symmetry requirements are satisfied for higher order connections.

In the simple propositional case, logic clauses take the form $A_1, A_2, ..., A_n \leftarrow B_1, B_2, ..., B_m.$ which says that (A_1 or A_2 or …. or A_n) if (B_1 and B_2 and … and B_n); they are program clauses if $n = 1$ and $m \geq 0$: we can have rules e.g. $A \leftarrow B, C.$ saying $A \vee \neg(B \wedge C) \equiv A \vee \bar{B} \vee \bar{C}$, and assertions e.g. $D \leftarrow .$ saying that D is true.

A logic program consists of a set of program clauses and is activated by an initial goal statement. In Conjunctive Normal Form (CNF), the clauses contain one positive literal. Basically, logic programming in Hopfield model can be treated as a problem in combinatorial optimization. Therefore it can be carried out in a neural network to obtain the desired solution. Our objective is to find a set of interpretation (i.e., truth values for the atoms in the clauses which satisfy the clauses (which yields all the clauses true). In other words, we want to find 'models'.

The following algorithm shows how a logic program can be done in a Hopfield network based on Wan Abdullah's method (Sathasivam, 2015):

i) Given a logic program, translate all the clauses in the logic program into basic Boolean algebraic form.
ii) Identify a neuron to each ground neuron.
iii) Initialize all connections strengths to zero.
iv)Derive a cost function that is associated with the negation of all the clauses, such that $\frac{1}{2}(1 + S_x)$ represents the logical value of a neuron *X*, where S_x is the neuron corresponding to

X. The value of S_x is define in such a way that it carries the values of 1 if X is true and -1 if X is false. Negation (neuron X does not occur) is represented by $\frac{1}{2}(1-S_x)$; a conjunction logical connective is represented by multiplication whereas a disjunction connective is represented by addition.

v) Obtain the values of connection strengths by comparing the cost function with the energy, H.

vi) Let the neural networks evolve until minimum energy is reached. Checked whether the solution obtained is a global solution.

We do not provide a detail review regarding neural network logic programming in this paper, but instead refer the interested reader to Wan Abdullah (Wan Abdullah, 1993).

4. New Activation Function

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 (1999) proposed a new activation function as followed:

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

$$V_{X_i} = \frac{\tanh\left(\frac{x_0}{u_0}\right)+0.5\left(1+\tanh\left(\frac{U_{X_i}-x_o}{u_o}\right)\right)}{1+\tanh\left(\frac{x_0}{u_0}\right)} \quad (U_{X_i} \geq 0) \tag{7}$$

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.

5. 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 new activation function) had been 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 (Sathasivam and Pei Fen, 2013). So, by using this approach we can get a clear visualization on doing logic programming in HONN.

6. Simulation And Discussion

Firstly, we generate random program clauses. Then, we initialize initial states for the neurons in the clauses. Next, we let the network evolves until minimum energy is reached. We test the final state obtained whether it is a stable state. If the states remain unchanged for five runs, then we consider it as stable state. Following this, we calculate corresponding final energy for the stable state. If the different between the final energy and the global minimum energy is within tolerance value, then we consider the solution as global solution. Then, we calculate ratio of global solutions.

We run the relaxation for 1000 trials and 100 combinations of neurons so as to reduce statistical error. The selected tolerance value is 0.001. All these values are obtained by try and error technique, where we tried several values as tolerance values, and selected the value which gives better performance than other values .

From the ABM we developed 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 (NC1, NC2, NC3) for the new activation function compare with the sigmoid function. It can be observed that when the network gets larger or more complex, the new activation function seems to perform better and continuously compared with the sigmoid function. This is due to the capacity of the activation function which is higher than sigmoid function. Bu using the new activation function neurons are able to relax to global minima values rather than stuck in local minima values. While using sigmoid function, the neurons get stuck and unable to jump the energy barrier to relax in global states. From the global minima graphs, when the neurons relaxed to global solutions the ratio will approached near to 1. These shows that the new activation function do perform well.

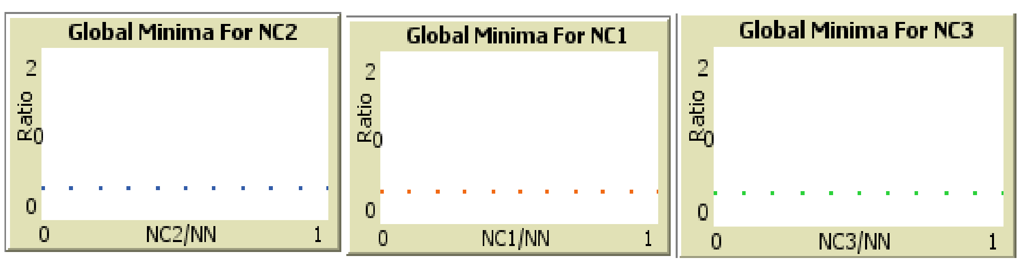

Figure 2: Global Minima Ratio for NC1, NC2 and NC3

7. Conclusion

From the study found that the ability of new activation function in doing in logic program on Hopfield network is better than Wan Abdullah method .It provides a better result in term of global minima ratio. Agent based modeling that had been carried out verified the validity of the new activation function performance compare Wan Abdullah method. This completes our illustration of computer simulation to test the validity and strength of the proposed method of doing logic programming in Hopfield network.

Acknowledgement

This research is partly financed by Incentive Grant Scheme (1001/ PMATHS/823032) by Universiti Sains Malaysia.

REFERENCES

Hopfield, J.J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities.(1982). *Proceedings. Natl. Acad. Sci. USA.* , **79(8)**, pp 2554-2558.

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

Hopfield, J.J. and Tank, D.W. Neural Computation of Decisions in Optimization Problems. (1985). *Biol. Cyber*, **52**, pp 141-152.

Saratha Sathasivam. (2015). Acceleration Technique for Neuro Symbolic Integration. Applied Mathematical Sciences,, 9, 9-12, 409-418.

Saratha Sathasivam and Ng Pei Fen. (2013). Developing Agent Based Modeling for Doing Logic Programming in Hopfield Network, Applied Mathematical Sciences, 7, 1, 23-35.

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

Wan Abdullah, W.A.T. The Logic of Neural Networks. (1993). *Phys. Lett. A.*, **176.**, pp 202-206.

Zeng, X. & Martinez, R. (1999). A new activation function in the Hopfield Network for Solving Optimization Problems. In Fourth International Conference On Artificial Neural Networks And Genetic Algorithms, pp 1-5.