Learning Rules Comparison in Neuro-Symbolic Integration

Saratha Sathasivam

***Abstract*—Pseudo inverse learning rule and hyperbolic activation function performance will be evaluated and compared with the sign constraint method and Hebb rule. Comparisons are made between these rules to see which rule is better or outperformed other rules in the aspects of computation time, memory and complexity. From the computer simulation that has been carried out, the hyperbolic activation function performs better than the other learning methods.**

***Index Terms*—Pseudo inverse, hebb rule, hyperbolic activation function, capacity**

I. Introduction

Neural network [1] 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 [2] 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 analyze the performance comparison of doing logic programming in Hopfield network by using different learning rules: Pseudo-inverse learning rule, hyperbolic activation function, sign constraint method and Hebb Rule. Part of the pleminary result of this paper had been published in [3, 4].

II. 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 [5, 6]. Part of this section has been reportedly earlier in some papers [7-10].

The Hopfield network [11] is a dynamic system. It can be written in its continuous form as:

$$\frac{du_i}{dt} = \sum_{j \neq i} w_{ij} s_j + I_i \qquad (1)$$
$$s_i = f\ u_i),$$

where u_i denotes the net function of neuron i, s_i represents the nonlinear output of neuron i, w_{ij} is the weight and I_i denotes an external input. The activation function f is usually either a hard limiter or a sigmoid. One notices that the output of each neuron is fed back to the input of all the other neurons except for its own input.

The Hopfield net can be completely described by a Lyapunov function E which is also referred to the Hopfield energy function:

$$E = -\frac{1}{2}\sum_i \sum_{j \neq i} s_i s_j w_{ij} - \sum_i s_i I_i \qquad (2)$$

One can easily verify that this is a Lyapunov function. One notices that $\frac{du_i}{dt} = -\frac{E}{ds_i}$. Hence, the energy function completely describes the Hopfield net. Since a Lyapunov function exists, we know that the Hopfield net is stable.

Assuming that the weight matrix is symmetric (w_{ij}=w_{ji}) and that the neurons are asynchronously updated, the Lyapunov function converges to a local minimum without oscillating between different states. That is, the state of the Hopfield net converges and the state corresponds to a local minimum of the Lyapunov function.

In general, a Hopfield net can either be utilized as an associate memory to store and retrieve information or to solve combinatorial optimization problems. In this report only the latter is addressed.

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 [7].

Logic programming can be seen as a problem in combinatorial optimization, which may therefore be carried out on a Hopfield neural network. The neurons are used to store the truth values of the atoms and writing a cost function which is minimized when all the clauses are satisfied. For logic programming, we chose an energy function

Manuscript received August 9, 2011
Saratha Sathasivam is with the School of Mathematical Sciences, Universiti Sains Malaysia, 11800 USM, Penang (Email: saratha@cs.usm.my)

representing the number of unsatisfied clauses, given a set of logical interpretations (i.e. neural state assignments) , or the sum of "inconsistencies", where the inconsistency of a particular clause is given as the Boolean value of the negation of that clause. The applied methodology may be summarized in the following way: given an optimization problem, find the cost function that describes it, design a Hopfield network whose energy function must reach (one of) its minima at the same point in configuration space as the cost function, so that the stable configurations of the network correspond to solutions of the problem. We do not provide a detail review regarding neural network logic programming in this paper, but instead refer the interested reader to Wan Abdullah [2].

III. Learning Rules

In this section, we will discuss the other learning rules: Hebbian rule, Pseudo-inverse rule and hyperbolic activation function.

A. Hebbian Rule

The Hebbian learning rule is given by:

$$w_{ij}^{v} = 0 \forall i, j \text{ and } w_{ij}^{v} = w_{ij}^{v-1} \frac{1}{N} \sum_{u} \xi_i^v \xi_j^v \tag{3}$$

where w^v is the state of weight matrix after the vth patterns have been learnt but before the $(v+1)th$ pattern has been introduced, ξ^v is the pattern for the v steps, where N is the number of neurons. The Hebbian learning rule is local and incremental, but has a low absolute storage capacity of $n/2In(n)$. Overall, its performance is poor in terms of storage capacity, attraction, and spurious memories.

B. Pseudo-inverse Rule

The pseudo-inverse rule is given by

$$w_{ij}^{v} = \frac{1}{N} \sum_{v=1}^{m} \sum_{u=1}^{m} \xi_i^v (Q^{-1})^{vu} \xi_j^u \tag{4}$$

where $Q = \frac{1}{N} \sum_{k=1}^{n} \xi_k^v \xi_k^u$, and m is the total number of patterns with $m < n$, N is the number of neurons.

The pseudo-inverse learning rule (PI), also known as projection learning rule. PI is accredited for its high retrieval capability. However, the PI is not local or incremental, because it involves the calculation of an inverse.

C. Hyperbolic Activation Function

The activation function in the Hopfield network is the sigmoid function. 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 [12] proposed a hyperbolic activation function as followed:

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

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.

D. Sign Constraint Method

Davey & Adams [13] investigated sign constrained for higher capacity in associative memory models. A possible difficulty with the normal neurons learning rule is that connection strengths can (and do) change sign during the learning process. This is not thought to happen and indeed Dale's rule [14] states that all different synapses from a given neuron are all either excitatory or inhibitory. For a neural network this is equivalent to requiring that all outgoing weights from a given unit have the same sign, and this cannot change over time.

A general sign constraint mechanism consists of a matrix of signs, $S_{ij} = \pm 1$, corresponding to each weight in the network, together with requirement that: $S_{ij} J_{ij} > 0$. The sign-bias of these weights is the ratio of positive to negative weights. It is defined as:

$$J_{ij}{}' = J_{ij} + \frac{S_i S_j}{N} \tag{6}$$

where J_{ij}' is the recent connection strengths after neuron's updating, S_i is the initial neuron's state , S_j is the neuron's final state (after updating procedure) and N is the number of neurons.

As is well known, normal neuron learning will converge on a solution, if one exists, since the connection strengths changes always move the weight vectors towards ones that embed the initial vectors.

IV. Theory Implemenntation

We generate random program clauses. Then, we initial states are initialize randomly for the neurons in the clauses. Next, we let the network relax until minimum energy is reached. We test the final state obtained whether it is a stable state where the states remain unchanged for more than five time steps, then we consider it as stable state.

Later, we calculate the corresponding final energy for the stable state. If the difference between the final energy and the global minimum energy is within a tolerance value (determine by the user), then we consider the solution as global solution. We measure the global solution ratio between the three types of learning rules: Hyperbolic Activation function, Hebbian learning , Sign Constraint Method Pseudo-inverse rule.

We run the relaxation for 100 trials and 1000 combinations of neurons so as to reduce statistical error. The selected tolerance value is 0.0001. All these values were obtained by trial and error.

V. Results and Discussion

We simulated the network for these three methods. The following Figure 1 shows the global minima ratio (Number of global minima solutions/Number of runs) using hyperbolic activation function approach.

It is clearly to see from the graph that the value of global minima ratio is decreasing when the number of neurons, NN is increased from 10 to 60. When the value of global minima ratio is increased, the effectiveness of the learning rule clearly seems to be increased also. However, from the result obtained, we can conclude that when the number of neurons, NN is increased simultaneously, the learning capacity using hyperbolic activation function is decreased. Thus, we can conclude that the effectiveness of this Method decreases as number of neurons, NN increases as shown in the Figure 1.

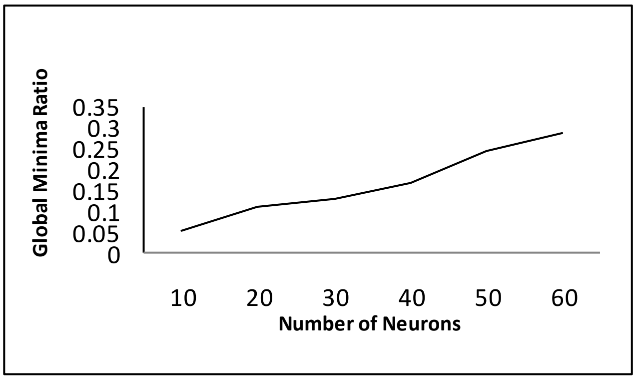

Fig. 1.Global Minima Ratio for Hyperbolic Activation Function method

The following Figure 2 shows the global minima ratio obtained by using Hebbian Rule.

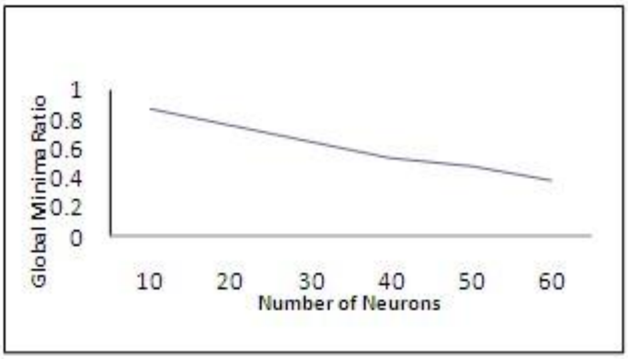

Fig. 2.Global Minima Ratio for Hebbian Learning Method

It is clear from the graph that the value of global minima ratio, is increasing when the value of number of neurons, NN is increased from 10 to 60. The higher the value of global minima ratio, the more effective is the learning rule in Hopfield network. We can observed that the effectiveness of Hebbian learning rule is increasing as shown in Figure 2 as the number of neurons increased. However, when we compare the result obtained with hyperbolic activation function method, we can conclude that for number of neurons, NN range from 10 to 60, the Hebbian learning rule is less effective than Wan Abdullah's method when the network get more complex.

The following figure shows the global minima ratio for Pseudo-inverse learning rule.

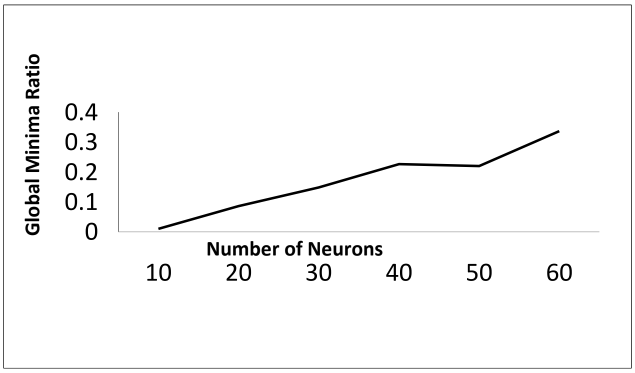

Fig. 3.Global Minima Ratio for Pseudo-inverse Method

From the graph, it can be observed that, by using the Pseudo inverse learning rule, the global minima ratio, increases when the number of neurons, NN increases. The higher the average global minima ratio, the more effective is the learning rule.

The following figure shows the global minima ratio for sign constraint method.

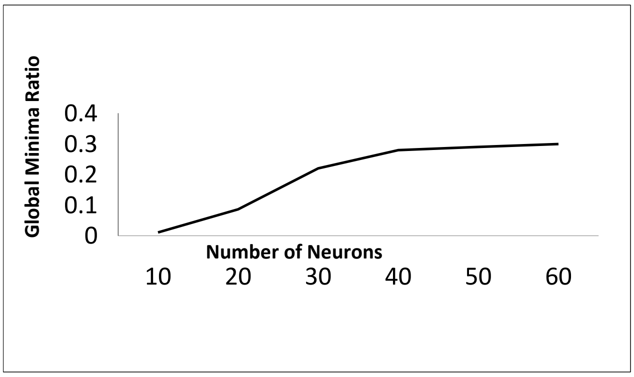

Fig. 4.Global Minima Ratio For Sign Constraint Method

By comparing the result obtained with other learning rule and pseudo inverse method, we notice that the global minima ratio and capacity of Pseudo inverse learning rule is better than other learning rule. When the network gets larger, we notice that the Pseudo inverse learning rule outperform other learning rules.

Thus, from here, we conclude that the performance of doing logic programming in Hopfield network can be accelerated mostly by using Pseudo Inverse learning rule, followed by Hyperbolic activation function, Sign Constraint method and finally Hebbian learning rule as the number of neurons, NN increases.

The results obtained here agreed with our theory. In hyperbolic activation function, we update the activation function to calculate the current state of neurons. So, the capacity for the comparison can accommodate higher number of neurons. Meanwhile for pseudo inverse rule involves inverse calculation. Due to that, the accuracy is higher, the performance increase. Lastly, the Hebbian rule, which is a primitive rule and fluctuate with the neurons states. Due to that, it is the most backward learning rule.

VI. CONCLUSION

The performance of the neuro-symbolic integration can be accelerated by using few learning rules. In this paper, we have discussed theoretically and validate the results by using computer simulation results. The performance had been accelerated in order by using Pseudo-inverse rule, hyperbolic activation function method, sign constraint method and Hebbian learning.

ACKNOWLEDGMENT

This research is partly financed by short term grant (304/PMATHS/6310066) and from Universiti Sains Malaysia.

REFERENCES

[1] Kaslik, E. & Balint, S, "Configuration of steady states for Hopfield-type neural networks" in Applied Mathematics and Computation, 182(1), 2006, pp 179-186.

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

[3] Saratha Sathasivam.2010.Usage of New Activation Function in Neuro-Symbolic Integration. In 4th Asian Physics Sympossium, Bali Indonesia.

[4] Saratha Sathasivam. 2010. Neuro symbolic integration using pseudo inverse rule. In Annual International Conference on Advance Topics in Artificial Intelligence. Phuket, Thailand. (ISBN: 978-981-08-7654-8), pp A-69-A-73.

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

[6] Browne, A & Sun. R., "Connectionist inference models" in *Neural Networks*, 14(10), 2001**,** pp1331–1355.

[7] Saratha Sathasivam. 2009. Enhancing Logic Programming Performance in Recurrent Hopfield Network, European Journal of Scientific Research, 37(1), pp 1-7.

[8] Saratha Sathasivam. 2009. Logical Content in the Recurrent Hopfield Network without Higher Order Connections, European Journal of Scientific Research, 37(3), pp 361-367.

[9] Saratha Sathasivam. 2009. Learning Rule Performance Comparison in Hopfield Network, American Journal of Scientific Research, Issue 6, pp 15-22.

[10] Saratha Sathsivam & Wan Abdullah, W.A.T. 2010. The Satisfiability Aspect of Logic on Little Hopfield Network, American Journal of Scientific Research, Issue 7, pp 90-105.

[11] Saratha Sathasivam. 2010. Upgrading Logic Programming in Hopfield Network, Sains Malaysiana, 39(1), pp115-118.

[12] Zeng, X. & Martinez, R. A new activation function in the Hopfield Network for Solving Optimization Problems. In Fourth International Conference on Artificial Neural Networks and Genetic Algorithms, 1999, pp 1-5.

[13] Davey, M and Adams, R.G. Sign Constrained High Capacity Associative Memory Models, Neural Computation: Springer, (2003).

[14] Dale, H.H. Pharmacology and Nerve Ending. In *Proceedings of the Royal Society of Medicine*, 28, pp 319-322, (1935)