

Mean Field Theory in Doing Logic Programming Using Hopfield Network

Muraly Velavan¹, Zainor Ridzuan bin Yahya¹, Mohamad Nazri bin Abdul Halif² & Saratha Sathasivam³

¹Institute of Engineering Mathematics, Universiti Malaysia Perlis, Malaysia

²School of Microelectronic Engineering, Universiti Malaysia Perlis, Malaysia

³School of Mathematical Sciences, Universiti Sains Malaysia Penang, Malaysia

Correspondence: Institute of Engineering Mathematics, Universiti Malaysia Perlis, 02600, Arau, Perlis, Malaysia.
E-mail: dsmuraly@yahoo.com

Received: February 27, 2015

Accepted: March 8, 2015

Online Published: December 20, 2015

doi:10.5539/mas.v10n1p154

URL: <http://dx.doi.org/10.5539/mas.v10n1p154>

The research is financed by Fundamental Research Grant Scheme (203/ PMATHS/6711368) by Ministry of Higher Education Malaysia and Universiti Sains Malaysia.

Abstract

Logic program and neural networks are two important perspectives in artificial intelligence. Logic describes connections among propositions. Moreover, logic must have descriptive symbolic tools to represent propositions. Meanwhile representation of neural networks on the other hand is in non-symbolic form. The objective in performing logic programming revolves around energy minimization is to reach the best global solutions. On the other hand, we usually gets local minima solutions also. In order to improve this, based on the Boltzmann machine concept, we will derive a learning algorithm in which time-consuming stochastic measurements of collerations are replaced by solutions to deterministic mean field theory (MFT) equations. The main idea of mean field algorithm is to replace the real unstable induced local field for each neuron in the network with its average local field value. Then, we build agent based modelling (ABM) by using Netlogo for this task.

Keywords: logic program, neural networks, mean field theory, agent based modelling

1. Introduction

The major domain of neuro-symbolic integration is designed by the theory are usually known as deductive systems which less such elements of human reasoning as adaptation, learning and self-organisation. Meanwhile, neural networks, known as a mathematical model of neurons in the human brain, and have various abilities, and moreover, they also provide parallel computations and therefore can perform some calculations quicker than classical learning algorithms. Hopfield network is a feedback (recurrent) neural network (Hopfield, 1982) designed by John Hopfield, consisting of a set of N interconnected neurons which each neurons are linked to all others in all the directions. It has synaptic strength pattern which involve Lyapunov function E (energy function) for energy minimization events. It operates as content addressable memory systems with binary or bipolar threshold units. More explanation will be carry out in section 2.

On the other hand, logic connected with true and false. Mean while in the logic programming, a set of Horn clauses that created by atoms are presented to determine the real values of each atom involves in the clauses. Neurons are used to store the truth value of atoms to derive a cost function for energy minimazation when all the clauses arefulfilled. Furthermore, a bi-direction mapping between propositional logic formulas and Lyapunov energy functions of a symmetric neural networks had been introduced by Gadi Pinkas (Pinkas, 1991) and Wan Abdullah (Wan Abdullah, 1991 & Wan Abdullah, 1992). The advantages of Wan Abdullah's method (also known as Direct Method) is it involves around propositional Horn clauses with learning capability of the Hopfield network which enable the network to hunt for the good solutions, when the corresponding clauses in the logic program are given, with the corresponding solutions may change as new clauses are added.

In this following paper, we are motivated to develop agent based modelling (ABM) for integration mean field theory for doing logic programming in the Hopfield network. An ABM is a new computational modelling

venture which is an analysing systems that representing the ‘agents’ that involving and simulating of their interactions. Their attributes and behaviours will be classified together through their interactions to become a scale.

The layout of the paper as follows. Section 2 consists an outline of Hopfield network and in section 3, we introduced a method of doing logic programming in Hopfield neural network. Mean while in section 3 and 4, mean field theory and agent based modelling will be discussed. Lastly, section 6 and 7 consists the simulation results and summarize remarks regarding the work proposed.

2. Hopfield Networks

Hopfield network [6] is illustrated in Figure 1 below. This network is mainly been used for solving combinatorial optimization problems and as a content addressable memory.

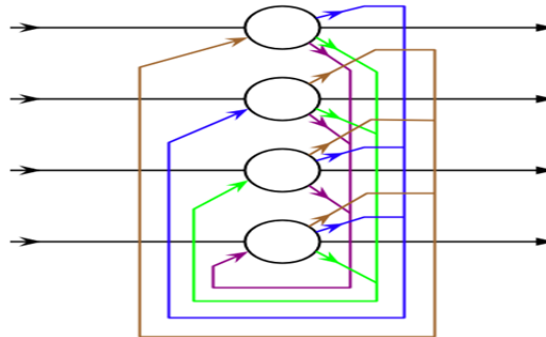


Figure 1. Discrete Hopfield network with four neurons

Hopfield network is mostly been used to solve combinatorial optimisation problem and content addressable memory problem (such as Travelling Salesman Problem). Besides this, Hopfield network also has few interesting features. One of the features is the distributed representation. Memory can be stored as a pattern and overlap upon one another different memories over the same set of processing elements. Besides that, it is also distributed and has asynchronous control (one neuron is updated in each cycle). Each processing neuron makes their own decisions corresponding to their local situation. It works as content addressable memory that patterns can be store and retrieve when needed. Finally, its fault tolerance criteria can let the networks still be functioning even although a few of processing elements had failed completely (Ding, 2010, Cheung, 1993 & Bologna, 2004).

Overall the Hopfield network involves of N neurons that are labeled as variable $x_i(t), i = 1, 2, \dots, N$. The

neurons are in bipolar, thereby activation value for variable can be stated as $x_i \in \{-1, 1\}$, comply with the

dynamics $x_i \rightarrow \text{sgn}(h_i)$ where sgn is known as the signum function. The induced local field is given by

$h_i = \sum_j w_{ij} x_j + w_i$ where w_i is the bias (negative of the threshold) is used externally to neuron j. w_{ij} is the connection strength from neuron i to j.

An energy function for the discrete Hopfield networks example for the second order Hopfield networks is given

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} x_i x_j - \sum_i w_i x_i \tag{1}$$

which the Lyapunov function is decrease monotonically with the network relaxation.

This model can be generalized for including higher order connections. This changes the ‘local field’ into

$$h_i = \left(\dots + \sum_j \sum_k w_{ijk} x_j x_k + \sum_j w_{ij} x_j + w_i \right) \tag{2}$$

where “.....” represents the higher orders, and an energy function is written as follows:

$$E = \dots - \frac{1}{3} \sum_i \sum_j \sum_k w_{ijk} x_i x_j x_k - \frac{1}{2} \sum_i \sum_j w_{ij} x_i x_j - \sum_i w_i x_i \quad (3)$$

where $J_{ijk} = J[ijk]$ for i, j, k distinct, with [...] representing permutations in cyclic order, and $J_{ijk} = 0$ for any i, j, k equal. The updating rule described as:

$$x_i(t+1) \rightarrow \text{sgn}[h_i(t)] \quad (4)$$

3. Logic Programming

Hopfield network (HN) is been used in the task of doing logic programming model due to its ability in solving constraint optimization problems. Mainly Hopfield network is used to minimize the logical inconsistency exists in the interpretations of logic programs and clauses. The procedure bellows shows how logic programming is carried out in Hopfield network by using Wan Abdullah’s method (Sathasivam, 2010 & Sathasivam, 2013).

i) Translate all the clauses involved in the given logic program into basic Boolean algebraic form. For an example, $D \leftarrow E, F$ as $D \vee \neg(E \wedge F) = D \vee \neg E \vee \neg F$.

ii) A neuron to each ground atom involve is set.

iii) All the connections strengths are initialized to zero (tabula rasa).

iv) A cost function that is related with the negation of all the clauses are derived. For an example $\frac{1}{2}(1 + S_D)$

represents the logical value of a neuron D , where S_D is the neuron corresponding to D . The value of S_D is interpreted in such a way that it carries the values of 1 if D is true and -1 if D is false. Negation (neuron D does

not occur) is denoted by, $\frac{1}{2}(1 - S_D)$; $E_p = \frac{1}{2}(1 - S_D) \frac{1}{2}(1 + S_E) \frac{1}{2}(1 + S_F) + \dots$ a conjunction logical

connective ‘and’ is denoted by multiplication. Mean while a disjunction connective ‘or’ is denoted by addition.

v) Calculate the synaptic strengths by equating the cost function with the energy function, H .

vi) The neural networks relaxed until minimum global energy. Hereby the neural states resemble a solution interpretation for the logic program, and the interpretation obtained are checked. If the interpretations fulfil the corresponding logic program then the solution obtained is classify as a global solution.

A model for a corresponding logic program can be found by doing logic programming in Hopfield network. I, to carry out a logic program, we need to build up a simulator to run it. So, we focused on agent based modelling. In order to increase the global minima, we introduce mean field theory (MFT). This technique is integrated in doing logic programming in Hopfield network to enhance the capability of the network.

4. Mean Field Theory

In a stochastic neuron, the firing mechanism is described by a probabilistic rule; Peterson and Anderson (Peterson and Anderson, 1988) have applied the mean field approximation to the Boltzmann machine with n units and derived the mean field equations:

$$\langle s_i \rangle = \tanh\left(\left(\sum_{j=1}^n W_{ij} \langle s_j \rangle + I_i\right)/T\right), i = 1, \dots, n \quad (5)$$

where $s = (s_1, \dots, s_n) \in \{-1, 1\}^n$ is the corresponding output value, W_{ij} is the weight of the synaptic strength from unit j to the unit i , I_i is the threshold of unit i , and $T(> 0)$ is known as the temperature parameter. In addition the weight matrix $W = (W_{ij})$ is a symmetric matrix which is the same as in the Hopfield model, and $\langle s_i \rangle$ represents the mean value of the output value s_i at the time the state of the Boltzmann machine converges to the Boltzmann distribution:

$$P(s) = \frac{1}{Z} \exp\left(-\frac{F(s)}{T}\right) \quad (6)$$

where $Z = \sum_{\{s\}} \exp(-F(s)/T)$ at a given temperature T . Hereby, we represent $\sum_{\{s\}}$ the total sum over all

configurations of the corresponding system.

Peterson and Anderson (Peterson and Anderson, 1988) also suggested the iterative method:

$$x_i(t + 1) = \tanh \left(\left(\sum_{j=1}^n W_{ij} x_j(t) + I_i \right) / T \right) \quad (i = 1, \dots, n), (t = 0, 1, 2, \dots) \quad (7)$$

where t represents discrete time as instants 0, 1, 2, This discrete-time recurrent neural network is known as the MFT neural network (MFT) and in this model every unit is updated at the same time (synchronous) and asynchronous where only the unit that is randomly selected among n units is updated. In this paper, we consider only the asynchronous MFT model in which only one unit selected cyclically is updated.

We may formally state that the approximation of the mean-field as the corresponding average of some function of a random variable is approximated as the function of the average of that corresponding random variable. Mainly the mean field approximation is to switch between the actual fluctuating induced local field for each neuron in the network with its average value.

$$v_j^{approx} = \langle v_j \rangle = \sum_i w_{ji} \langle x_i \rangle \quad (8)$$

Accordingly, we may compute the average state $\langle x_j \rangle$ for neuron j embedded in a stochastic machine made up of a total of N neurons as:

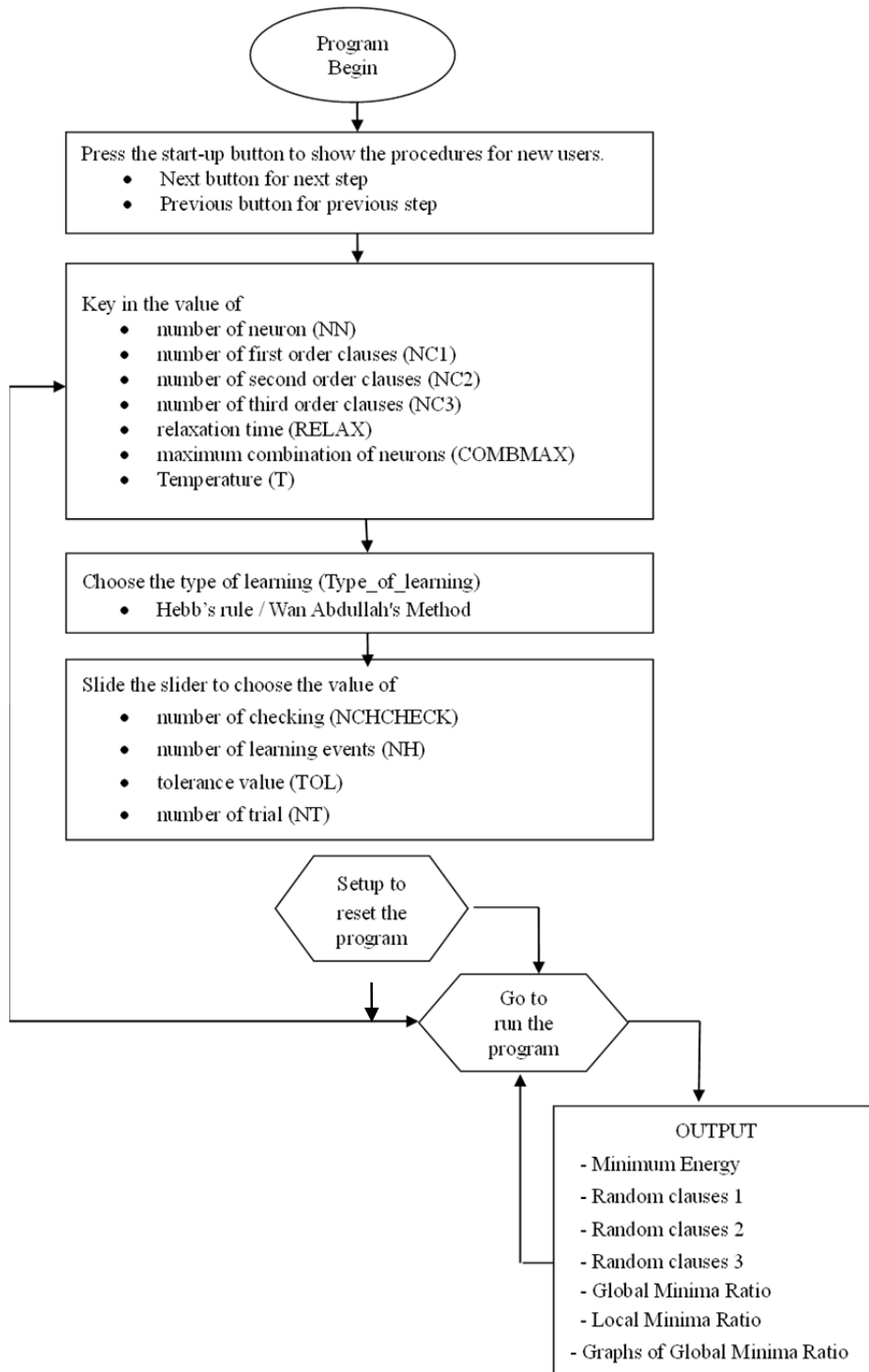
$$\langle x_j \rangle = \tanh \left(\frac{1}{2T} \sum_i w_{ji} \langle x_i \rangle \right) \quad (9)$$

So, in doing logic programming in Hopfield network, we change the steps in computing local field. Alternately we use equation (7)-equation (8) to compute the average local field which is more efficient in increasing the network computation in finding model for the corresponding logic program.

5. Agent Based Modelling

Initially, a simulator of Hopfield networks that is using a conventional type computer had been created instead of each time building up a new network design or storing a new set of memories. We used NETLOGO version 6.0 as the platform. It saves lots of computational complexity and time for the programmer to rebuild a new system from time to time. So that, a computer program which emulates exactly what the user want needs to be constructed in order to simulate the mechanism of Hopfield Network. It will be easier task for the programmer to modify the program and to store a new set of data. Thus, an agent based modelling had been designed for the user to run the simulator. In this paper, an agent based modelling which was implemented the mean field theory in doing logic programming in Hopfield network had been created.

Moreover, agent-based Modelling (ABM) which also named as 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 assembling together through their interactions to form 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 use. In addition, ABM reveals the appearance of the systems from low to high level outcomes and it make improvement by overcoming the traditional modelling limitations such as allowing agent learning and adaption, limited access to knowledge and access to information. So, by using this approach we can get a clear picture on doing logic programming by integrating mean field theory. The following flow chart shows implementation of the algorithm in agent based modelling.



6. Experimental Results and Discussion

Firstly, program clauses are generated randomly. The program is limited up to third order clauses for simplicity purpose. Then, we initial states are been initialized randomly for the neurons in the clauses. Then, we apply the procedure of doing logic programming in Hopfield network. To integrate MFT, we focused on average induced local field. Then, we applied Boltzmann machine and simulated annealing technique, to “push” the stuck neurons from local minima to global minima values by going through energy relaxation looping. Then the network is let to be evolve until minimum energy is reached. The final state obtained are been tested whether it is a stable state (states remain unchanged for more than five time steps). If the states are stable states, then the

corresponding final energy for the stable states are calculated. The solution is classified as a global solution if the difference between the final energy and the global minimum energy is within a tolerance value (determined by the user). Else, the solutions are classified as local solutions. The simulator is run for 100 trials and 100 combinations of neurons. The tolerance value is set to 0.001. All these values were obtained by trial and error mechanism. The following screen shot shows the results obtained.

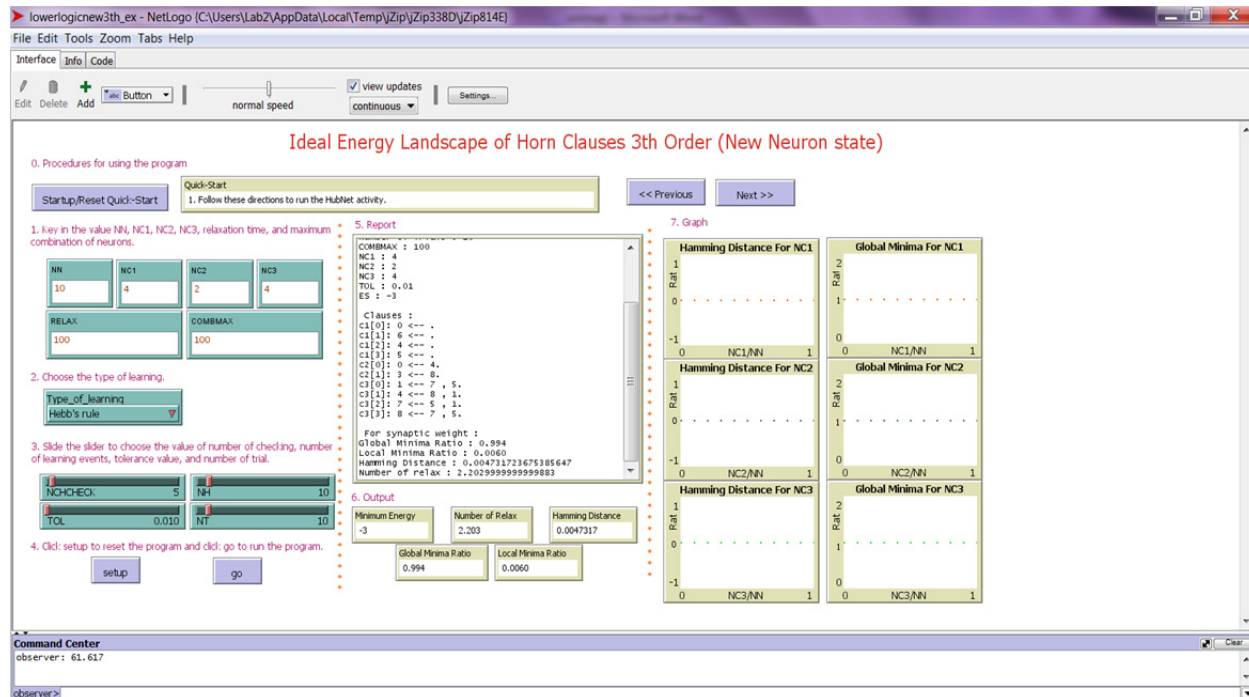


Figure 2. Screen shot of the results

We simulated the network for the mean field theory for doing logic programming in Hopfield network up to third order clauses. Due to all the results obtained are quite similar, so we just presented one of the results only to avoid overlapping. Figure 2 shows the global minima ratio (Number of global minima solutions/Number of runs) obtained by the integration of logic programming with MFT. From the screen shot obtained, 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(one literal per clause), NC2(two literals per clause), NC3(three literals per clause)). From the figure, we can see that when the network gets larger or more complex, the mean field theory manage to perform better with less computation time, 61.617 sec. This is due to the simulated annealing procedure carried out by the MFT where the neurons are forced to jump the energy barriers to relax into global solutions by varying the temperature. By using this method the neurons are able to relax to global minima values rather than stuck in local minima values. This mean field approximation often becomes exact in the limit of infinite range interactions, where each spin interacts with all the others. This is because the induced local field is then the sum of very many terms, and a central limit theorem can be applied.

7. Conclusion

Mean field theory implemented in carrying out logic programming in Hopfield network has been proven effective in accelerating the computational ability of neuro symbolic integration. Agent based modelling built based on this theory has verified the outperforming of MFT. The computer simulation using ABM also agrees with the proposed method.

Acknowledgement

The research is financed by Fundamental Research Grant Scheme (203/ PMATHS/6711368) by Ministry of Higher Education Malaysia and Universiti Sains Malaysia.

References

Brenton, C. (2002). Stability analysis of higher-order neural networks for combinatorial optimization.

International Journal of Neural Systems, 12, 177-186.

- Bologna, G. (2004). Is it worth generating rules from neural network ensembles? *Journal of Applied Logic*, 325-348. <http://dx.doi.org/10.1016/j.jal.2004.03.004>
- Cheung, K. W., Lee, T. (1993b). *Boundary Detection by Artificial Neural Network*. IEEE Proceeding of 1993 International Joint Conference on Neural Networks, Nagoya, 2, 1189-1194.
- Ding, Y., Dong, L., Wang, L., & Wu, G. (2010a). A High Order Neural Network to Solve Crossbar Switch Problem. In: Wong, K.W. et al. (eds.) *ICONIP 2010, Part II, LNCS 6444*, Heidelberg: Springer pp. 692–699.
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings National Academy of Science USA*, 79, 2554-2558. <http://dx.doi.org/10.1073/pnas.79.8.2554>
- Peterson, C., & Anderson, J. (1988). A Mean Field Theory Learning Algorithm for Neural Networks. *Complex System 1*, 995-1019.
- Pinkas, G. (1991). Energy minimization and the satisfiability of propositional calculus. *Neural Computation*, 3, 282-291.
- Pinkas, G. (1991). *Propositional no monotonic reasoning and inconsistency in symmetric neural networks*. Proceedings of the 12th International Joint Conference on Artificial Intelligence, 525-530.
- Saratha, S., Ng, P. F., & Muraly, V. (2013). *Boltzmann Machine and Reverse Analysis Method*. IEEE 7th International Power Engineering and Optimization Conference (PEOCO 2013), Malaysia, 52-56. <http://dx.doi.org/10.1109/peoco.2013.6564514>
- Sathasivam, S. & Wan, A. T. (2010). The Satisfiability Aspect of Logic on Little Hopfield Network. *ISSN 1450-223X (7)*, 90-105.
- Wan Abdullah, W. A. T. (1992). Logic Programming on a Neural Network. *Int .J. Intelligent Sys*, 7, 513-519. <http://dx.doi.org/10.1002/int.4550070604>
- Wan Abdullah, W. A. T. (1991). Neural Network Logic. Benhar, O., Bosio, C., del Giudice, P., & Tabet, E. (eds.), *Neural Networks: From Biology to High Energy Physics*, ETS Editrice, Pisa, pp.135-142
- Wen, P. H. (2008). A Review of Hopfield Neural Networks for Solving Mathematical Programming Problems, *European Journal of Operational research*, 198(3), 675-688. <http://dx.doi.org/10.1016/j.ejor.2008.11.002>

Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).