Sains Malaysiana 49(11)(2020): 2847-2857
http://dx.doi.org/10.17576/jsm-2020-4911-23

Novel Random *k* Satisfiability for $k \leq 2$ in Hopfield Neural Network

(Novel Rawak *k* Kepuasan untuk $k \leq 2$ dalam Rangkaian Neural Hopfield)

Saratha Sathasivam, Mohd. Asyraf Mansor*, Ahmad Izani Md Ismail, Siti Zulaikha Mohd Jamaludin, Mohd Shareduwan Mohd Kasihmuddin & Mustafa Mamat

ABSTRACT

The k *Satisfiability logic representation (*k*SAT) contains valuable information that can be represented in terms of variables. This paper investigates the use of a particular non-systematic logical rule namely Random* k *Satisfiability (RAN*k*SAT). RAN*k*SAT contains a series of satisfiable clauses but the structure of the formula is determined randomly by the user. In the present study, RAN*k*SAT representation is successfully implemented in Hopfield Neural Network (HNN) by obtaining the optimal synaptic weights. We focus on the different regimes for* $k \leq 2$ *by taking advantage of the non-redundant logical structure, thus obtaining the final neuron state that minimizes the cost function. We also simulate the performances of RAN*k*SAT logical rule using several performance metrics. The simulated results suggest that the RAN*k*SAT representation can be embedded optimally in HNN and that the proposed method can retrieve the optimal final state.*

Keywords: Artificial neural network; Hopfield neural network; logic programming; random satisfiability

ABSTRAK

Perwakilan logik k *Kepuasan mengandungi maklumat berguna yang diwakilkan dalam sebutan pemboleh ubah. Kajian ini mengkaji penggunaan suatu peraturan logik yang tidak sistematik iaitu logik* k *Kepuasan Rawak (RAN*k*SAT). RAN*k*SAT mengandungi siri klausa penuh tetapi struktur rumusnya ditentukan secara rawak oleh pengguna. Dalam kajian ini, perwakilan RAN*k*SAT berjaya dilaksanakan untuk Rangkaian Neural Hopfield (HNN) dengan memperoleh pemberat sinapsis yang optimum. Fokus diberikan kepada rejim berbeza bagi* $k \leq 2$ *dengan menggunakan struktur logik yang tidak berulang dan justeru memperoleh secara optimal keadaan neuron akhir yang meminimumkan fungsi kos. Prestasi logik* k *Kepuasan Rawak disimulasi dengan menggunakan beberapa indikator prestasi tertentu. Keputusan simulasi menunjukkan perwakilan RAN*k*SAT boleh dimasukkan secara optimum dalam HNN dan teknik yang telah dicadangkan berupaya memperoleh semula perwakilan neuron akhir yang optimum.*

Kata kunci: Kepuasan rawak; rangkaian neural buatan; rangkaian neural Hopfield; pengaturcaraan logik

Introduction

Artificial Neural Networks (ANNs) are computing models that can be applied to identify and resolve a specific problem according to the appropriate objective function and patterns. These capabilities allow ANNs to be considered as techniques for solving large categories of optimization problems including scheduling problems (Sharma & Garg 2020), complex valued analysis (Kobayashi 2019) and health management (Liu et al. 2020). Generally, conventional ANN are expected to generate an implicit, qualitative, and predictive model that solves a problem with high accuracy in the fastest possible time. In layman terms, ANN acquire the learnable behaviour based on the experience (previous data), generate the learned pattern (synaptic weight) and solves the problem via the learned pattern. An early and simple ANN is Hopfield Neural Network (HNN). HNN is a simple ANN that capitalizes on the importance of the input and output layer in finding the optimal synaptic weight (Hopfield & Tank 1985). HNN has been shown to be useful in various types of optimization problems such as games (Fitzsimmons & Kunz 2020), and image encryption (Wang & Li 2019). Despite the recent and rapid development in HNN, minimal attention has been given to represent the output of HNN in the form of symbolic rules.

Representing the problems of non-deterministic polynomial time by transforming them to a propositional satisfiability (SAT) logical rule has been a successful strategy to address various industrial issues involving constraint Satisfaction problem. In order to develop an integrated approach, a number of scholars have established the SAT as a universal language for logic and reasoning and declarative logic. The main question that fascinates researchers is: how neural networks behave according to the propositional logic? This is because this paradigm attempts to represent the output of the neural networks using symbolic system. The combination of logic programming into ANN must take into account the synaptic weight structure which will result in the correct neuron updates. Pinkas (1991) asserts that the connectionist ANN model with symmetric weight can be represented as a Quadratic Energy Function. The minimization of the energy corresponds to the possible solution of the constraint optimization problem. The first attempt to incorporate propositional logical rule with neural network was by Abdullah (1992). In this paper, the logical rule is embedded into Hopfield Neural Network (HNN) by comparing the cost function with the Lyapunov energy function. The target of creating an improved version of logic programming in HNN was continued by Sathasivam (2010). In this paper, systematic relaxation method is implemented to reduce unnecessary neuron oscillation. In another development, Hamadneh et al. (2012) proposed logic programming in Radial Basis Function Neural Network (RBFNN). The proposed RBFNN obtained the output weight by minimizing the error metric between the output layer and the hidden layer. Information about this network can be found in Hamadneh et al. (2012) and Mansor et al. (2020). Regarding the logical structure, several developments stemmed from the idea of creating systematic logical rule. Kasihmuddin et al. (2017) and Mansor et al. (2017a) proposed the first systematic logical rule namely *k*SAT into HNN. The proposed methods managed to retrieve the correct final state up to 90% of the time. These findings inspired other applications to incorporate *k*SAT in HNN. Several simulated applications such as Very Large-Scale Integration (Mansor et al. 2016a), Bezier curve reconstruction (Kasihmuddin et al. 2016), Pattern Satisfiability (Mansor et al. 2016b). The robustness of the proposed method has been extended to logic mining via *k* Satisfiability Reverse Analysis Method (*k*SATRA). The proposed logic mining attracted several applications such as e-sport (Kho et al. 2020), disease screening (Kasihmuddin et al. 2018a), social media (Mansor et al. 2018) and student performance (Kasihmuddin et al. 2019a). It is worth pointing out that the proposed logic mining achieves an acceptable range of accuracy. Since then, systematic logical rule became the main impetus for the development of several *k*SAT variant such as Maximum *k* Satisfiability (Kasihmuddin et al. 2018b) and randomized 2 Satisfiability (Mansor et al. 2020). However, there has not been much effort to implement non-systematic logical rule into neural network. Non-systematic logical rule provides flexibility for the real dataset to be represented by the neural network.

The study of random instance in Boolean SAT has been a major research focus in recent years due the random input structure that contribute to only 2 output states. Maneva and Sinclair (2008) proposed 3SAT that consist of random instances. This study illustrates the use of random selection that constitute the 3SAT formula by successfully implemented possible assignment with the defined threshold. The random structure in SAT has been beneficial for several applications. Gao (2009) has reported interesting random structure of Weighted SAT that capitalize several form of SAT formulas. In this work, the proposed threshold in SAT contributed toward the development of the data reduction. Worth mentioning that, several interesting studies that implemented Random notion can be found in Amendola et al. (2020), Fan and Shen (2011), and Schawe et al. (2019). The common ground of all of the mentioned studies are the use of Random SAT to represent their case. The main idea of the implementation of Random SAT is the flexibility to represent the number of literals that is not limited to only *k* variable per clause. To the best of our knowledge, there is no recent integration of Random *k*SAT as a logical rule in ANN. In our study, *k*SAT is extended to non-systematic logical rule by incorporating random structure into the logical formula. The underlying assumption of the proposed logical rule is to represent the formula in a random and flexible manner. Thus, the contributions of the current paper are: A novel Random *k* Satisfiability (RAN*k*SAT) is proposed by implementing the random structure that involves first and second order Satisfiability logical rule. Implementation of RAN*k*SAT into HNN by creating a learning phase that minimizes the cost function of the HNN. A comprehensive analysis of the RAN*k*SAT for both learning and retrieval phase. An effective HNN model incorporating the new logical rule was constructed and the proposed network is seen to be beneficial in finding the correct approximate solution for various non-deterministic problem such as scheduling, control theory and function approximation.

The rest of the paper is organized as follows: The novel non-systematic logical rule namely Random *k* Satisfiability (RAN*k*SAT) are given in the next section. The proposed RAN*k*SAT will be implemented in HNN in the subsequent section. The methods and experimental setup will be given in the next two sections. The simulation of the RAN*k*SAT HNN will be discussed thoroughly after that. Finally, concluding remarks are given in last section.

MATERIALS AND METHODS

THE PROPOSED RANDOM *K* SATISFIABILITY (RAN*K*SAT)

Random k Satisfiability (RAN*k*SAT) is a logical representation which consists of a non-systematic number of literal per clause. RAN*k*SAT is a variant of Boolean formula that usually represented in Conjunctive Normal Form (CNF) where each clause contains random number of variables. The general equation for RAN*k*SAT is as follows

$$P_{RANkSAT} = \bigwedge_{i=0}^{n} C_i^{(2)} \bigwedge_{i=0}^{m} C_i^{(1)} \tag{1}$$

$n, m \in \mathbb{Z}^{+}$ and the definition of the clause in $P_{RANkSAT}$ is given by:

$$C_i^{(k)} = \begin{cases} \left(A_i^* \vee B_i^*\right), k = 2 \\ D_i^* \quad , k = 1 \end{cases} \tag{2}$$

where $A_i^* \in \{A_i, \neg A_i\}$, $B_i^* \in \{B_i, \neg B_i\}$ and $D_i^* \in \{D_i, \neg D_i\}$. The first and second order clause are given as $C_i^{(1)}$ and $C_i^{(2)}$, respectively. The choice of variable (Positive or Negative literal) is determined randomly. In this case, F_r is a Conjunctive Normal Form (CNF) formula where the clauses are chosen uniformly, independently without replacement among all $2^r \binom{m+n}{r}$ non-trivial clause of length r. Note that, A_i^* exists in the $C_i^{(k)}$, if the $C_i^{(k)}$ contains either A_i or $\neg A_i$ and the mapping of $V(F_r) \rightarrow \{-1,1\}$ is called logical interpretation. According to Kho et al. (2020), the Boolean value for the mapping is defined as 1 (TRUE) and -1 (FALSE). One of the examples for RAN*k*SAT formulation is:

$$P_{RANkSAT} = \left(A_1 \vee \neg B_1\right) \wedge \left(\neg A_2 \vee B_2\right) \wedge \neg D_1 \tag{3}$$

According to (3), $P_{RANkSAT}$ consist of $C_1^{(2)} = \left(A_1 \vee \neg B_1\right)$, $C_2^{(2)} = \left(\neg A_2 \vee B_2\right)$ and $C_1^{(1)} = \neg D_1$. Thus, the solutions for each of the outcomes in (3) could be diversified as more combinations of solution correspond to different form of $P_{RANkSAT}$. Note that, the outcome of (3) is $P_{RANkSAT} = -1$ if $\left(A_1, A_2, B_1, B_2, D_1\right) = (1,1,1,1,1)$ with 2 clauses are satisfied $\left(C_1^{(2)}, C_2^{(1)}\right)$. Additionally, $P_{RANkSAT} = -1$ if $\left(A_1, A_2, B_1, B_2, D_1\right) = (-1,-1,-1,-1,1)$ with 2 clauses are satisfied $\left(C_1^{(2)}, C_2^{(1)}\right)$ is also the alternative solution of attaining the condition of the stated $P_{RANkSAT}$ in (3). In this paper, we limit our investigation to $k \leq 2$ by considering only first and second order logical rule.

RAN*k*SAT IN HOPFIELD NEURAL NETWORK

The prominent structure of Hopfield Neural Network (HNN) involves the interconnected bipolar neurons without the intervention of hidden neurons (Hopfield & Tank 1985). The synaptic weights are strictly symmetrical, without self-mapping among the respective neurons. Due to the capability of Content Addressable Memory (CAM), it is considered as the dynamic storage system for the synaptic weights (Sathasivam 2010). Given an initial vector that is mapped to the neuron state $S_i = (S_1, S_2, S_3, ..., S_N)$ and without any assistance of noise, the HNN will converge to the equilibrium that correponds to the nearest minimized H_p (Barra et al. 2018). Hence, the final state of the HNN corresponds to the solution of the combinatorial problem. The neurons in HNN are considered bipolar, $S_i \in (-1,1)$ conform to the dynamics $S_i \rightarrow \operatorname{sgn}(h_i)$. The general asynchronous updating rule of HNN is given by:

$$S_i(t) = \begin{cases} 1, \text{ if } \sum_{j}^{N} W_{ij} S_j(t) + \beta \geq 0 \\ -1, \text{otherwise} \end{cases} \tag{4}$$

where W_{ij} describes the synaptic weight matrix of HNN that establishes the strength of the connections from neuron j to i with pre-determined bias β. In this work, the HNN is adopted as the central network in $P_{RANkSAT}$ programming. The formalism of logic programming in HNN does not impose any restriction on the accepted type of clauses as long as the proposed propositional logic is satisfiable (Abdullah 1996). $P_{RANkSAT}$ can be embedded into HNN by assigning each variable into $m+n$ neuron. The firing structure of the neuron are based on the defined cost function. Hence, the cost function $E_{P_{RANkSAT}}$ that governs the combinations of HNN and $P_{RANkSAT}$ is:

$$E_{P_{RANkSAT}} = \sum_{i=1}^{NC} \prod_{j=1}^{m+n} T_{ij} \tag{5}$$

where NC and $m+n$ are the number of clauses and the number variables in $P_{RANkSAT}$ respectively. Note that, the inconsistency of $P_{RANkSAT}$ is given as:

$$T_{ij} = \begin{cases} \frac{1}{2}\left(1 - S_A\right), \text{ if } \neg A \\ \frac{1}{2}\left(1 + S_A\right), \text{ otherwise} \end{cases} \tag{6}$$

Note that, the value of $E_{P_{RANkSAT}}$ is proportional to the number of 'inconsistencies' of the clause $\left(C_i^k = -1\right)$. The more C_i^k that is unsatisfied, the bigger the value of $E_{P_{RANkSAT}}$. Minimum $E_{P_{RANkSAT}}$ corresponds to the 'most consistent' selection of S_i. Hence, the updating rule for $P_{RANkSAT}$ in

HNN is defined as:

$$h(t)=\sum_{j=1,i\neq j}^{m+n} W_{ij}^{(2)} S_j(t)+W_i^{(1)} \tag{7}$$

$$S_i(t)=\begin{cases} 1\ ,\ h(t)\geq 0 \\ -1,\ h(t)<0 \end{cases} \tag{8}$$

where $W_{ij}^{(2)}$ and $W_i^{(1)}$ are second and first order synaptic weights of the embedded $P_{RANkSAT}$. Equations (7) and (8) are important to ensure the neurons S_i will always converge to $E_{RANkSAT} \to 0$. In order to evaluate the quality of the retrieved S_i, we utilize the Lyapunov energy function, $H_{P_{RANkSAT}}$, defined as:

$$H_{P_{RANkSAT}}=-\frac{1}{2}\sum_{i=1,i\neq j}^{m+n}\sum_{j=1,i\neq j}^{m+n} W_{ij}^{(2)} S_i S_j-\sum_{i=1,i\neq j}^{m+n} W_{ij}^{(1)} S_j \tag{9}$$

One of the properties of (9) is that energy portrayed from the $P_{RANkSAT}$ always decreases monotonically. The value of $H_{P_{RANkSAT}}$ signifies the value of the energy with respect to the absolute final energy $H_{P_{RANkSAT}}^{\min}$ obtained from $P_{RANkSAT}$ (Abdullah 1992). The value of $H_{P_{RANkSAT}}^{\min}$ can be further calculated by using the following equation:

$$H_{P_{RANkSAT}}^{\min}=-\left(\frac{\theta+2\eta}{4}\right) \tag{10}$$

where $\theta=n\left(C_i^{(2)}\right)$ and $\eta=n\left(C_i^{(1)}\right)$ that corresponds to $P_{RANkSAT}$. Hence, the quality of the final neuron state can be properly examined by checking the following condition:

$$\left|H_{P_{RANkSAT}}-H_{P_{RANkSAT}}^{\min}\right|\leq\partial \tag{11}$$

where ∂ is the tolorence value pre-determined by the user. Note that, if the embedded $P_{RANkSAT}$ does not satisfy (11), the final state obtained is trapped in local minimum solution. It should be mentioned that, $W_{ij}^{(2)}$ and $W_i^{(1)}$ can be effectively obtained by using Wan Abdullah method (Abdullah 1992). Hebbian learning has been reported to produce oscillating neuron state that will result in sub optimal value of $H_{P_{RANkSAT}}$. In this paper, the implementation of $P_{RANkSAT}$ in HNN is denoted as HNN-RANkSAT. Figure 1 shows the schematic diagram for NN-RANkSAT.

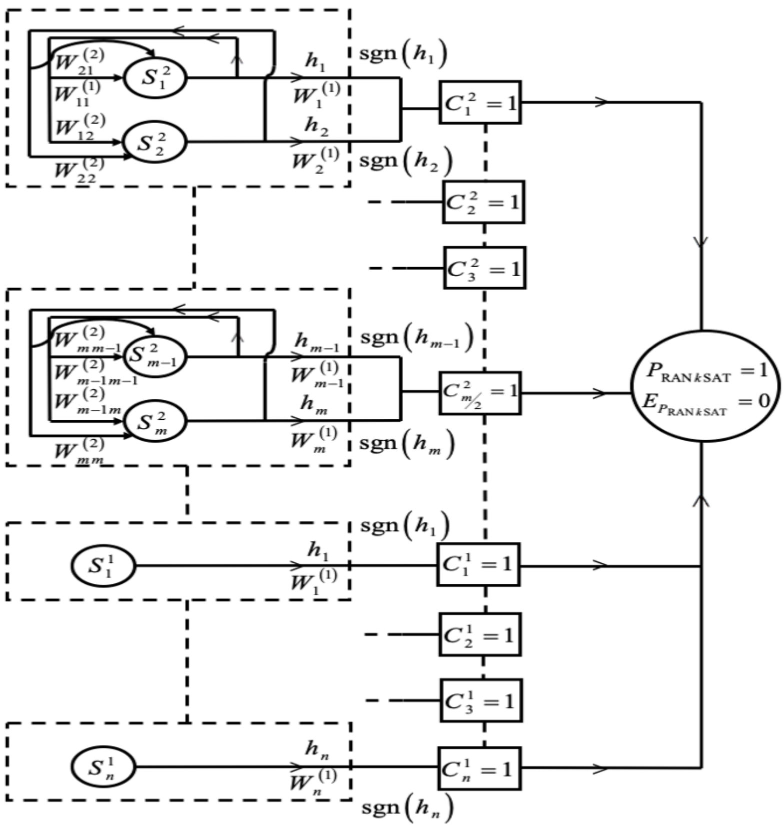

FIGURE 1. Schematic diagram for HNN-RANkSAT

SIMULATION STUDY

A simulation study was conducted to evaluate the performance of HNN model in analysing simulated datasets. The simulation was conducted in DEV C++ Version 5.11 in Windows 10, using an Intel Core i3 with 1.7 GHz processor. In order to make a fair comparison, HNN model was terminated after being executed more than the threshold computation time (24 h). The main task of the HNN model is to retrieve the neuron state that corresponds to the proposed $P_{RANkSAT}$. The parameters involved for the HNN-RAN*k*SAT are listed in Table 1.

TABLE 1. List of parameters for HNN-RAN*k*SAT

Parameter	Parameter value
$n\left(C_i^{(1)}+C_i^{(2)}\right)$	$10 \le C_i^{(1)}+C_i^{(2)} \le 20$
Neuron Combination (z)	100
Tolerance Value (∂)	0.001
$S_i^{(1)}$ selection	Random
$S_i^{(2)}$ selection	Random
$C_i^{(1)}$ selection	Random
$C_i^{(2)}$ selection	Random
No_Neuron String	100
Relaxation	3 (Sathasivam 2010)
Selection_Rate	0.2
Activation Function	Hyperbolic Activation Function (Mansor & Sathasivam 2016)

The global minima ratio (Zm) can be defined as the ratio between the number of global minimum solution with the total number of solution (Kasihmuddin et al. 2019b). In this case, if the HNN-RAN*k*SAT produces 10000 final state, the maximum value for Zm is 1. In terms of energy profile, global minimum energy can be assumed as the absolute minimum energy that corresponds to $E_{P_{RANkSAT}}=0$. The definition of Zm is:

$$Zm=\frac{1}{tz}\sum_{i=1}^{tz}N_{H_{P_{RANkSAT}}} \tag{12}$$

where z, t, and $N_{W_{G_{RANkSAT}}}$ represent the neuron combination, the number of trials and the global minimum energy of the proposed model, respectively. For example, the value of Zm = 0.9 signifies 90% of the final state is achieved the optimal final state. Another performance metric that examines the efficiency of the proposed HNN is Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). RMSE measures the error deviation between the optimal fitness and the current fitness of the HNN model. Higher RMSE value signifies higher error deviation of the network during learning phase of HNN. In this paper, the RMSE value of the learning phase is:

$$RMSE=\sum_{i=1}^{NC}\sqrt{\frac{1}{n}\left(f_{NC}-f_i\right)^2} \tag{13}$$

where f_{NC} and f_i are optimal fitness and current fitness, respectively. In addition, MAE measures the absolute differences between the optimal fitness and the current fitness of the HNN model. The equation for MEA is:

$$MAE=\sum_{i=1}^{NC}\sqrt{\frac{1}{n}\left|f_{NC}-f_i\right|} \tag{14}$$

Note that, $f_{NC} \neq 0$ because the probability of all neuron state to be -1 for one single initialization is almost zero. The complete implementation of the HNN-RAN*k*SAT is demonstrated in Figure 2.

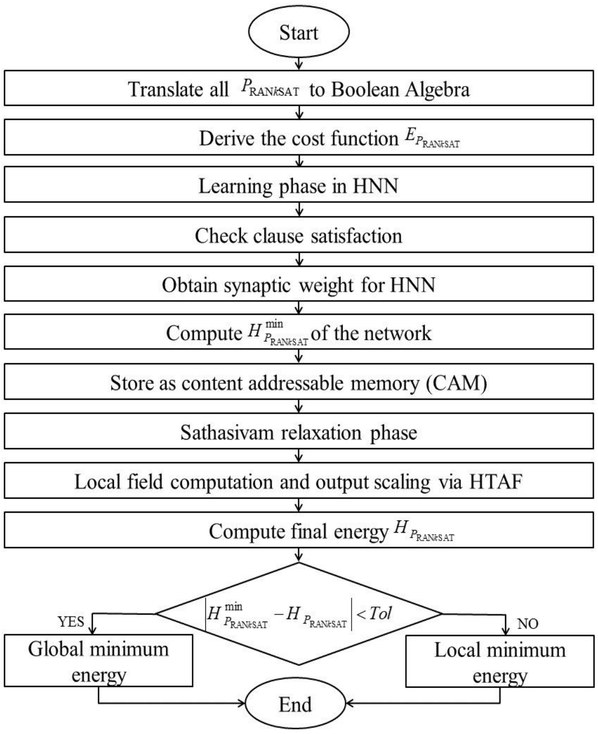

FIGURE 2. Implementation of HNN-RAN*k*SAT

Results and Discussion

The results of this study can be summarized into two main findings; the success of $P_{RANkSAT}$ learned by HNN, thus creating an optimal HNN-RAN*k*SAT, and the capability of HNN-RAN*k*SAT to portray the behaviour of $P_{RANkSAT}$ in a random environment. The synaptic weight of the HNN-RAN*k*SAT is shown in Figures 3 and 4. The values of synaptic weight were computed based on the cost

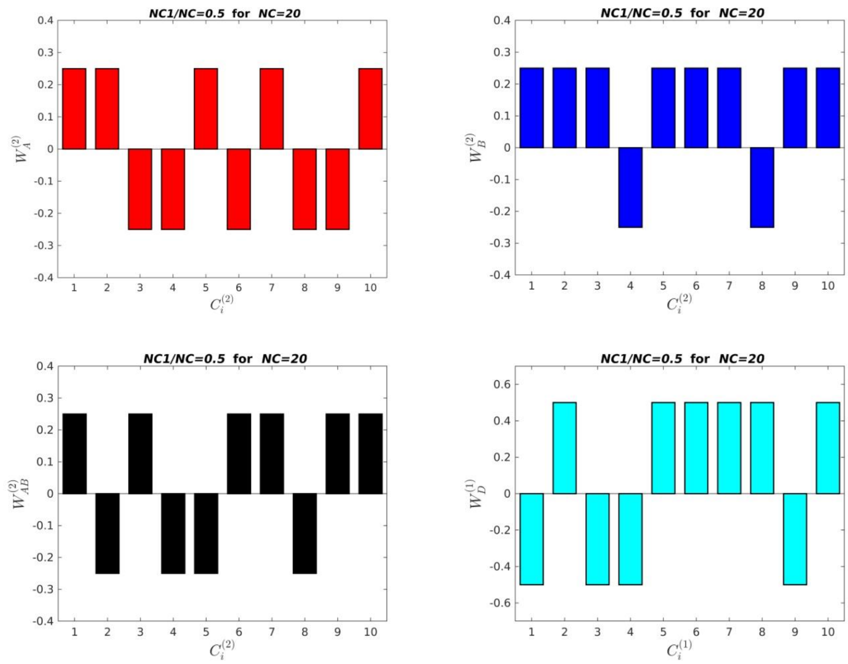

FIGURE 3. Synaptic weight for instance i =19 z = 100

function, $E_{P_{RANkSAT}}$ formed by the RAN*k*SAT with respect to the unique inconsistencies of $P_{RANkSAT}$ as shown in (6). Based on Figures 3 and 4, for the case of $P_{RANkSAT}$ the magnitude of the synaptic weight can be different but remain symmetrical with when $E_{P_{RANkSAT}}$ being compared to (9) via Wan Abdullah method (Abdullah 1992). However, the main focus of the analysis in Figures 3 and 4 is to determine the positive and negative literal tendencies of the $P_{RANkSAT}$. These graphs show the synaptic weight obtained during the learning phase for both $C_i^{(1)}$ and $C_i^{(2)}$ when 50% of $C_i^{(2)}$ is included into HNN-RAN*k*SAT. Note that, $W_B^{(2)}$ for instance $i = 19$ in Figure 3, has a higher tendency for positive literal B compared to negative literal $\neg D$. Similar observation can be deduced in Figure 4.

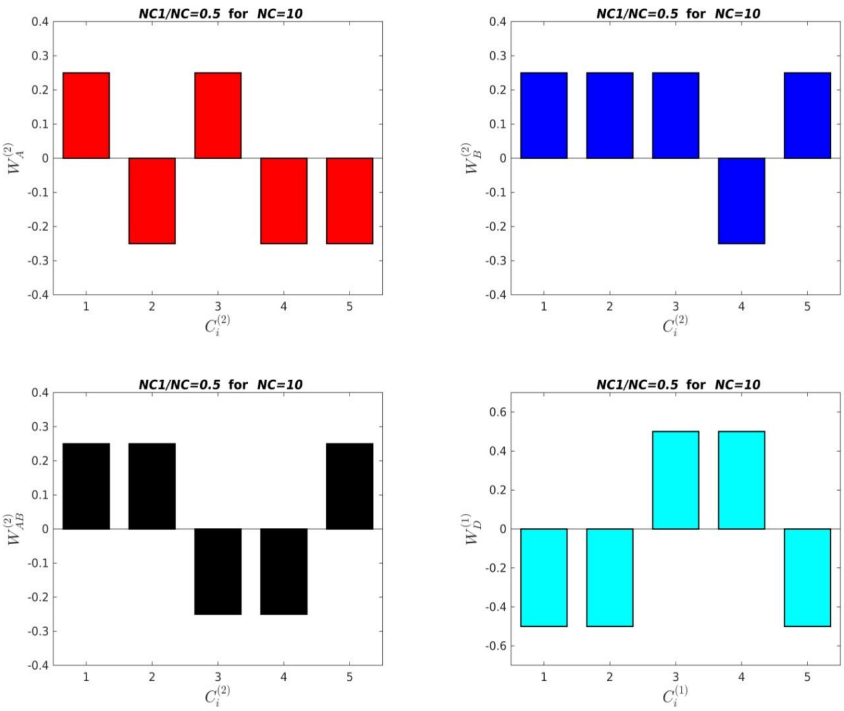

FIGURE 4. Synaptic weight for instance i =19 z = 100

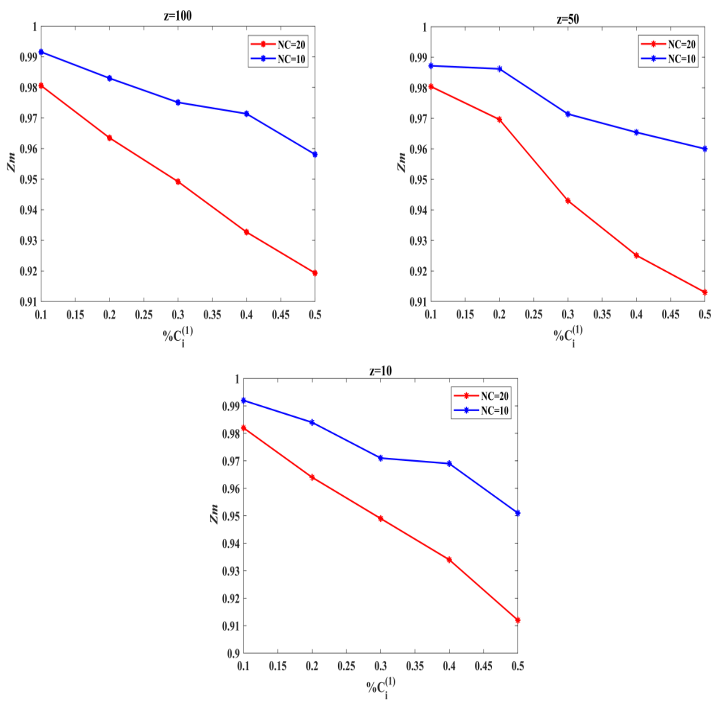

FIGURE 5. *Zm* evaluation for HNN-RAN*k*SAT *at* $z = 10, 50, 100$

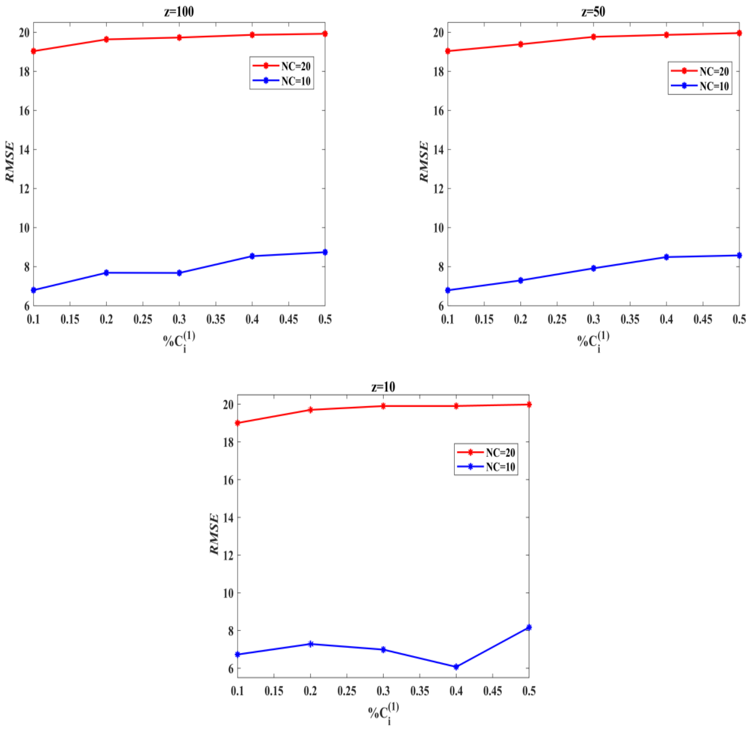

FIGURE 6. RMSE evaluation for HNN-RAN*k*SAT *at* z = 10, 50, 100

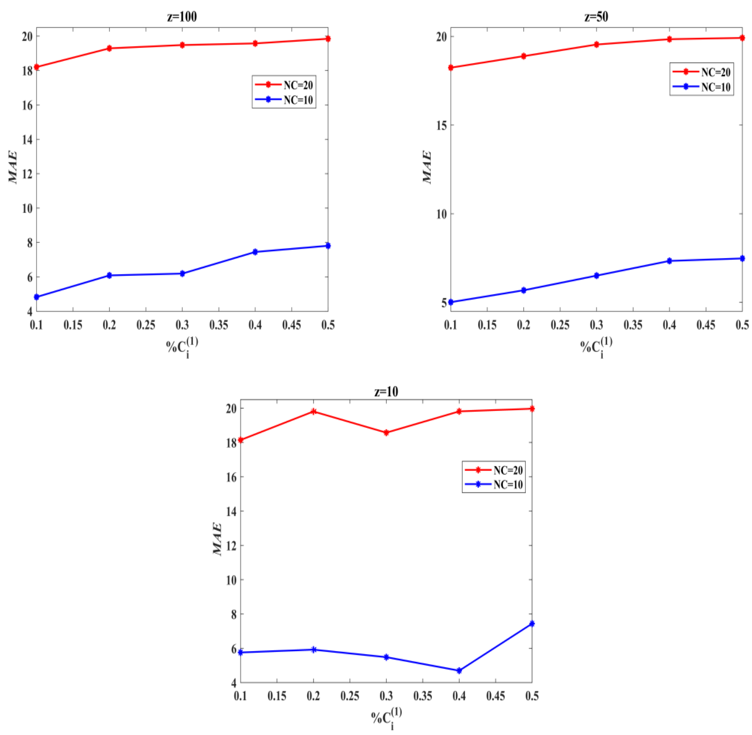

FIGURE 7. MAE evaluation for HNN-RAN*k*SAT *at* z = 10, 50, 100

Figures 5 to 7 represent the performance of HNN-RAN*k*SAT in terms of *Zm*, RMSE and MAE respectively. Based on Figures 6 and 7, the RMSE and MAE values for HNN-RAN*k*SAT were reported to increase as the number of neuron increase. For instance, at $\%C_i^{(1)} = 0.5$ about 60% difference in the RMSE and MAE evaluation between $NC = 10$ as compared to $NC = 20$ during learning phase of HNN-RAN*k*SAT. According to the RMSE and MAE value during learning phase, the proposed method manages to achieve $E_{P_{RANkSAT}} = 0$ for all the maximum combination z despite high learning error as the number of neuron increases. The random structure of the $P_{RANkSAT}$ increases the logical variation during the learning phase. $C_i^{(2)}$ clause is observed to reduce the complexity of the $P_{RANkSAT}$ due to the dimension flexibility of the logical rule. In other words, $C_i^{(2)}$ is relatively feasible to achieve $E_{P_{RANkSAT}} = 0$ compared to $C_i^{(1)}$. Although the Satisfiability of $C_i^{(1)}$ is guaranteed, the nature of exhaustive search will make it difficult for the HNN-RAN*k*SAT to learn the inconsistent interpretation $\neg P_{RANkSAT}$. It is fair to report that $P_{RANkSAT}$ is expected to deal with higher learning complexity compared to Maximum Satisfiability (Mansor et al. 2017b), 2 Satisfiability (Kasihmuddin et al. 2017), HornSAT (Sathasivam 2010) and 3 Satisfiability (Mansor & Sathasivam 2016). This is because the mentioned works only deal with systematic logical rule that is relatively easy to be learned.

Next, we discuss the retrieval capability. The retrieval capability can be assessed by analyzing the value of Zm as manifested in Figure 5. In general, the Zm was reported to significantly reduce especially within $0.3 \leq \%C_i^{(1)} \leq 0.5$. This is due to more possible neuron oscilation during the retrieval phase as the number of neuron increases. Although the value of Zm decreased as the number of neuron increase, HNN-RAN*k*SAT retrieved almost 90% global minimum solution $(Zm \to 1)$ for all cases of z. The proposed HNN-RAN*k*SAT has a good agreement with the study of Kasihmuddin et al. (2018) where the Zm obtained is approaching 1. The final neuron state of the HNN-RAN*k*SAT exhibit the $P_{RANkSAT}$ behaviour by retrieving the final state that is corresponds to the condition in (11). It is seen in Figure 7 that the number of global minimum energy decrease as the number of neuron increase. In this case, the neuron state that corresponds to the behaviour of $P_{RANkSAT}$ can be visualized as a basin of attraction in the energy profile. Thus, $H_{P_{RANkSAT}}$ that does not satisfy condition (11) is said to trapped in suboptimal solution. Despite the possible suboptimal behaviour of $P_{RANkSAT}$, HNN-RAN*k*SAT always converges to the nearest minimum solution. Under these circumstances, stochastic methods such as simulated annealing (Grabust et al. 2019) can be expected to drive the final state from the local minimum to the global minimum energy. The optimal relaxation time step is shown to increase the accuracy of HNN-RAN*k*SAT in retrieving more consistent final neuron state. The ease of optimization task (from suboptimal to optimal solution) is being called creativity of the HNN (Abdullah 1992). Therefore, the result obtained from the simulation illustrates that HNN-RAN*k*SAT achieved the optimal storage capacity which implies good fault tolerance property. In addition, the simulation only evaluates the retrieval capability of the HNN-RAN*k*SAT. Metaheuristics algorithm such as Artificial Bee Colony and Artificial Immune System can be expected to optimize the learning phase of HNN-RAN*k*SAT. From another perspective, the proposed $P_{RANkSAT}$ does not include the redundant logical variable because the Satisfiability of the redundant logical rule is not guaranteed.

Conclusion

Based on the results and discussion in the aforementioned section, some conclusions can be made from the study. First, a novel Random *k* Satisfiability logical rule namely RAN*k*SAT has been developed by combining first and second order logical rule that is Satisfiable in nature. The proposed RAN*k*SAT is structurally flexible and capitalize on the benefit of another logical rule such as 2 Satisfiability $k = 2$. Secondly, to the best of our knowledge, this is the first attempt to represent symbolic output of the HNN in terms of non-systematic logical rule. The proposed HNN-RAN*k*SAT is functionally different from the existing systematic logical rule (Kasihmuddin et al. 2018; Mansor et al. 2016; Sathasivam 2010). In light of this new logical rule, we tested the capability HNN-RAN*k*SAT in doing simulated datasets. The results of the simulation studies show that RAN*k*SAT can be embedded into HNN by minimizing the cost function that corresponds to the inconsistencies of the network. We also discuss the performance of the HNN-RAN*k*SAT in several performance metrics.

Some interesting open questions arise. For instance, what is the behaviour of the HNN-RAN*k*SAT when $k \leq 3$? This requires the use of other established logical rules such as 3 Satisfiability and Horn Satisfiability. Extending from that premise, the learning phase of HNN-RAN*k*SAT, Metaheuristics Algorithm such as Binary Artificial Bee Colony (Jia et al. 2014) and Binary Whale Optimization (Reddy et al. 2019) are possible techniques in reducing the learning complexity. These works are currently in progress and thus are deferred to later articles.

Acknowledgements

This research is supported by the Research University Grant Scheme by the Universiti Sains Malaysia (1001/PMATHS/8011131).

REFERENCES

Abdullah, W.A.T.W. 1996. Logic programming in neural networks. *International Journal of Computer Science* 9(1): 1-5.

Abdullah, W.A.T.W. 1992. Logic programming on a neural network. *International Journal of Intelligent Systems* 7(6): 513-519.

Amendola, G., Ricca, F. & Truszczynski, M. 2020. New models for generating hard random boolean formulas and disjunctive logic programs. *Artificial Intelligence* 279: 103185.

Barra, A., Beccaria, M. & Fachechi, A. 2018. A new mechanical approach to handle generalized Hopfield neural networks. *Neural Networks* 106: 205-222.

Fan, Y. & Shen, J. 2011. On the phase transitions of random *k*-constraint satisfaction problems. *Artificial Intelligence* 175(3-4): 914-927.

Fitzsimmons, M. & Kunze, H. 2019. Combining Hopfield neural networks, with applications to grid-based mathematics puzzles. *Neural Networks* 118: 81-89.

Gao, Y. 2009. Data reductions, fixed parameter tractability, and random weighted *d*-CNF satisfiability. *Artificial Intelligence* 173(14): 1343-1366.

Hamadneh, N., Sathasivam, S. & Choon, O.H. 2012. Higher order logic programming in radial basis function neural network. *Applied Mathematical Sciences* 6(3): 115-127.

Hopfield, J.J. & Tank, D.W. 1985. "Neural" computation of decisions in optimization problems. *Biological Cybernetics* 52(3): 141-152.

Jia, D., Duan, X. & Khan, M.K. 2014. Binary artificial bee colony optimization using bitwise operation. *Computers & Industrial Engineering* 76: 360-365.

Kasihmuddin, M.S.M., Mansor, M.A. & Sathasivam, S. 2019a. Students' performance via satisfiability reverse analysis method with Hopfield Neural Network. *Proceedings of the International Conference on Mathematical Sciences and Technology 2018 (MATHTECH2018): Innovative Technologies for Mathematics & Mathematics for Technological Innovation.* p. 060035.

Kasihmuddin, M.S.M., Mansor, M.A., Basir, M., Faisal, M. & Sathasivam, S., 2019b. Discrete mutation Hopfield neural network in propositional satisfiability. *Mathematics* 7(11): 1133.

Kasihmuddin, M.S.M., Mansor, M.A. & Sathasivam, S. 2018a. Satisfiability based reverse analysis method in diabetes detection. In *Proceedings of the 25th National Symposium on Mathematical Sciences (SKSM25): Mathematical Sciences as the Core of Intellectual Excellence*. p. 020020.

Kasihmuddin, M.S.M., Mansor, M.A. & Sathasivam, S. 2018b. Discrete Hopfield Neural Network in restricted maximum k-satisfiability logic programming. *Sains Malaysiana* 47(6): 1327-1335.

Kasihmuddin, M.S.M., Mansor, M.A. & Sathasivam, S. 2017. Hybrid genetic algorithm in the hopfield network for logic satisfiability problem. *Pertanika Journal of Science & Technology* 25(1): 139-152.

Kasihmuddin, M.S.M., Mansor, M.A. & Sathasivam, S. 2016. Bezier Curves Satisfiability Model in Enhanced Hopfield Network. *International Journal of Intelligent Systems and Applications* 8(12): 9-17.

Kho, L.C., Kasihmuddin, M.S.M., Mansor, M. & Sathasivam, S. 2020. Logic mining in league of legends. *Pertanika Journal of Science & Technology* 28(1): 211-225.

Kobayashi, M. 2019. Storage capacity of hyperbolic Hopfield neural networks. *Neurocomputing* 369: 185-190.

Liu, D., Wu, Y.L., Li, X. & Qi, L. 2020. Medi-Care AI: Predicting medications from billing codes via robust recurrent neural networks. *Neural Networks* 124: 109-116.

Maneva, E. & Alistair, S. 2008. On the satisfiability threshold and clustering of solutions of random 3-SAT formulas. *Theoretical Computer Science* 407(1-3): 359-369.

Mansor, M.A. & Sathasivam, S. 2016. Accelerating activation function for 3-satisfiability logic programming. *International Journal of Intelligent Systems and Applications* 8(10): 44-50.

Mansor, M.A., Jamaludin, S.Z.M., Kasihmuddin, M.S.M., Alzaeemi, S.A., Basir, M.F.M. & Sathasivam, S. 2020. Systematic Boolean satisfiability programming in radial basis function neural network. *Processes* 8(2): 214.

Mansor, M.A., Sathasivam, S. & Kasihmuddin, M.S.M. 2018. Artificial immune system algorithm with neural network approach for social media performance metrics. In *Proceedings of the 25th National Symposium on Mathematical Sciences (SKSM25): Mathematical Sciences as the Core of Intellectual Excellence*. p. 020072.

Mansor, M.A., Kasihmuddin, M.S.M. & Sathasivam, S. 2017a. Artificial immune system paradigm in the Hopfield network for 3-satisfiability problem. *Pertanika Journal of Science & Technology* 25(4): 1173-1188.

Mansor, M.A., Kasihmuddin, M.S.M. & Sathasivam, S. 2017b. Robust artificial immune system in the Hopfield network for maximum k-satisfiability. *International Journal of Intelligent Systems and Applications* 4(4): 63-71.

Mansor, M.A., Kasihmuddin, M.S.M. & Sathasivam, S. 2016a. VLSI circuit configuration using satisfiability logic in Hopfield network. *International Journal of Intelligent Systems and Applications* 8(9): 22-29.

Mansor, M.A., Kasihmuddin, M.S.M. & Sathasivam, S. 2016b. Enhanced Hopfield Network for pattern satisfiability optimization. *International Journal of Intelligent Systems and Applications* 8(11): 27-33.

Pinkas, G. 1991. Symmetric neural networks and propositional logic satisfiability. *Neural Computation* 3(2): 282-291.

Reddy, K.S., Panwar, L., Panigrahi, B.K. & Kumar, R. 2019. Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets. *Engineering Optimization* 51(3): 369-389.

Sathasivam, S. 2010. Upgrading logic programming in Hopfield network. *Sains Malaysiana* 39(1): 115-118.

Schawe, H., Bleim, R. & Hartmann, A.K. 2019. Phase transitions of the typical algorithmic complexity of the random satisfiability problem studied with linear programming. *PLoS ONE* 14(4): e0215309.

Sharma, M. & Garg, R. 2020. An artificial neural network based approach for energy efficient task scheduling in cloud data centers. *Sustainable Computing: Informatics and Systems* 26: 100373.

Wang, X.Y. & Li, Z.M. 2019. A color image encryption algorithm based on Hopfield chaotic neural network. *Optics and Lasers in Engineering* 115: 107-118.

Saratha Sathasivam, Ahmad Izani Md Ismail, Siti Zulaikha Mohd Jamaludin & Mohd Shareduwan Mohd Kasihmuddin
School of Mathematical Sciences
Universiti Sains Malaysia
11800 USM, Pulau Pinang
Malaysia

Mohd. Asyraf Mansor*
School of Distance Education
Universiti Sains Malaysia
11800 USM, Pulau Pinang
Malaysia

Mustafa Mamat
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin
21300 UniSZA Kuala Terengganu, Terengganu Darul Iman
Malaysia

*Corresponding author; email: asyrafman@usm.my

Received: 23 March 2020
Accepted: 18 May 2020