Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 42 (2014) 222 – 227
Procedia Computer Science

International Conference on Robot PRIDE 2013-2014 - Medical and Rehabilitation Robotics and Instrumentation, ConfPRIDE 2013-2014

Fault Diagnosis Method for Mobile Ad-Hoc Network by using smart Neural Networks

Maryam Ghiasi[a], Abbas Karimi*[a]
[a] Department of Computer Engineering, Faculty of Engineering, Arak Branch, Islamic Azad University, Arak, Markazi, Iran

Abstract

MANETs are dynamic collection of autonomous nodes that communicate with each other via wireless connections. One of the events that the network should have expected it to be a fault, and the behavior is more important, in this network. So that fault diagnosis can effect on final performance of the network in such a way that it does not fall under the negative impact of the fault. A non-linear neural network is a statistical method for modeling data or the tools to make decisions. Artificial neural network is a method for pattern recognition and classification. Error detection is a problem of categorization or classification. The use of neural networks can be useful in fault diagnosis in MANETs because of fault diagnosis is a classification problem. But one problem with this method is placed in a local optimum. Here a method based on the combination of the back-propagation algorithm, a local search algorithm and learning automata as efficient global search, is proposed. In the proposed method, the algorithm of learning automata adjusting learning rate on neural network according to given formula. For training and testing the neural network of the mobile network parameters that were measured, were used as input and output. The results show that the proposed method in terms of repeatability, reliability and lack of placement in a local optimum is better.

Peer-review under responsibility of the Center for Humanoid Robots and Bio-Sensing (HuRoBs)

Keywords: Neural Networks; Learning Automata; MANETs; Fault Diagnosis Systems

1. Introduction

Mobile ad hoc networks (MANETs) are self-organizing wireless networks, which can provide communication service in situations where infrastructure is either not available, not trusted, or should not be relied on in times of emergency[1,2].

Fault identification is one of the important part in many protocols. When the actual behavior is deviated by system or nodes of the system, a diagnosis function started to determine which node performed abnormal behavior that is called diagnosis[3]. Diagnosis is classified based on the occurrence of fault. Several diagnosis methods have been adopted based either on invalidation models, such as the PMC model, or comparison models, broadcast

* Corresponding author. Tel. +98-916-3113863.
E-mail address: Akarimi@iau-arak.ac.ir.

Peer-review under responsibility of the Center for Humanoid Robots and Bio-Sensing (HuRoBs)
doi:10.1016/j.procs.2014.11.055

comparison model and the generalized comparison model[4]. The comparison model is most promising approaching which a set of task is assigned to nodes and outcomes are compared with their neighbor's outcomes. Various generalized comparison approach have been used.

Three types of diagnosis models have been studied in the context of system-level self-diagnosis: testing, comparison, and probabilistic models. Testing models, such as the classical PMC model[5,6], and its variations such as the BGM model[7] and the HK models[8], assume that each node is assigned a subset of the other units to test and the diagnosis is based on the collection of test outcomes. While, comparison models, such as the Malek's comparison model[9] and the Chwa-Hakimi comparison model[10], assume that a set of jobs is assigned to pairs of distinct units, and the results are compared. The outcomes of these comparisons, i.e., the matching and mismatching results, are used as a basis in order to identify the set of faulty nodes. Fault diagnosis under the testing and comparison models assumes in general a worst case behavior. Finally, probabilistic models[2] do not assume any bound, but instead, only fault sets that have a no negligible probability of occurrence are considered. In this paper, we consider the comparison-based diagnosis approach. The comparison approach has been introduced independently by Malek[9] and by Chwa and Hakimi[10] giving rise to two models. The Malek's model is known as the asymmetric comparison model and that of Chwa and Hakimi is called the symmetric comparison model. In both models it is assumed that two fault-free nodes give matching (0) results while a faulty and a fault-free node give mismatching (1) outcomes. The two models differ in the assumption on tests involving a pair of faulty nodes. In the symmetric model, both test outcomes (0=1) are possible in this case, while in the asymmetric model two faulty nodes always give mismatching outputs (1).

2. The Proposed Model

2.1. Backpropagation Algorithm

Error backpropagation training algorithm, which is an iterative gradient descent algorithm, is a simple way to train multilayer feed forward neural networks.5 The BP algorithm is based on the gradient descent rule:

$$W(\mathrm{n}+1) = \mathrm{W(n)} + \eta G(\mathrm{n}) + \alpha[\mathrm{W(n)} - \mathrm{W(n-1)}] \tag{1}$$

Where W is the weight vector, n is the iteration number, η is the learning rate, α is the momentum factor, and G is the gradient of error function that is given by:

$$G(\mathrm{n}) = -\nabla\, \mathrm{E}_p(\mathrm{n}) \tag{2}$$

E_p is the sum of squared error given by:

$$W(\mathrm{n}+1) = \mathrm{W(n)} + \eta G(\mathrm{n}) + \alpha[\mathrm{W(n)} - \mathrm{W(n-1)}] \tag{3}$$

Where Tp,j and Op,j are desired and actual outputs for pattern p at output node j. A major problem encountered during implementation of the BP learning rule is proper choice and update of the learning rate η to allow convergence, while keeping the number of required iterations at a reasonable number. One of the main reasons for investigating the possibility of the adaptive learning rate rule is the desire to reduce the sensitivity of the learning on the learning rate, without adding more tuning parameters.

2.2. Learning Automata

Learning automata (LA) can be classified into two main families, fixed and variable structure learning automata. Examples of the FSLA are Tsetline, Krinsky, TsetlineG, and Krylov automata. A fixed structure learning automaton is a quintuple (α, Φ, β, F, G) where:

- $\alpha = (\alpha_1, ..., \alpha_R)$ is the set of actions that it must choose from.
- $\Phi = (\Phi_1, ... , \Phi_s)$ is the set of states.
- $\beta = \{0, 1\}$ is the set of inputs where 1 represents a penalty and 0 represents a reward.
- F: $\Phi\times\beta\rightarrow \Phi$ is a map called the transition map. It defines the transition of the state of the automaton on receiving input, F may be stochastic.
- G: $\Phi\rightarrow \alpha$ is the output map and determines the action taken by the automaton if it is in state Φ_j.

Variable-structure Automata: Variable-structure automaton is represented by sextuple $\langle \beta, \Phi, \alpha, P, G, T \rangle$ where β a set of inputs actions, Φ is a set of internal states, α a set of outputs, P denotes the state probability vector governing the choice of the state at each stage k, G is the output mapping, and T is the learning algorithm. The learning algorithm is a recurrence relation and is used to modify the state probability vector[10,11,12,13,14,15,16].

It is evident that the crucial factor affecting the performance of the variable structure learning automata is learning algorithm for updating the action probabilities. Various learning algorithms have been reported in the literature. In linear reward-penalty algorithm (L_{R-P}) scheme, the recurrence equation for updating p is defined as

$$p_j(\mathrm{k}+1) = \begin{cases} p_j(\mathrm{k}) + \mathrm{a}[1 - p_j(\mathrm{k})] & \mathrm{j} = i \\ (1-\mathrm{a})\, p_j(\mathrm{k}) & \forall j \neq i \end{cases} \tag{4}$$

if β(k) = 0 and

$$p_j(\mathrm{k}+1) = \begin{cases} (1-\mathrm{b})\,\mathrm{p_j(k)} & \mathrm{j} = \mathrm{i} \\ (\dfrac{1}{r-2}) + (1-\mathrm{b})\,\mathrm{p_j(k)} & \forall j \neq i \end{cases} \tag{5}$$

if β(k) = 1.

2.3. Neural Network equipped with Learning Automata

In all of the existing schemes, one automata have been associated to the network. The learning automata based on the observation of the random response of the neural network, adapted one of BP parameters. The interconnection of learning automata and neural network is shown in Fig. 1. Note that the neural network is the environment for the learning automata. The learning automata according to the amount of the error received from neural network adjust the parameter of BP algorithm. The actions of the automata correspond to the values of the parameters being calculated and input to the automata is some functions of the error in the output of neural network. A function of error between the desired and the actual outputs of network is considered as the response of the environment. Here we adjust learning rate in neural network with learning automata as below:

$\eta(t)=c^- .\eta(t-1)$: decrease learning rate or first action (α_1)
$\eta(t)=c^+ .\eta(t-1)$: increase learning rate or second action (α_2)
Where c^- is contraction coefficient and c^+ is growth coefficient and η is learning rate in neural network. After several experiment we select 0.5 for c^- and 1.05 for c^+. At first the probability of selection each action for L_{R-P} equal to 0.5, because r or number of actions equals to 2 (p=1/2).

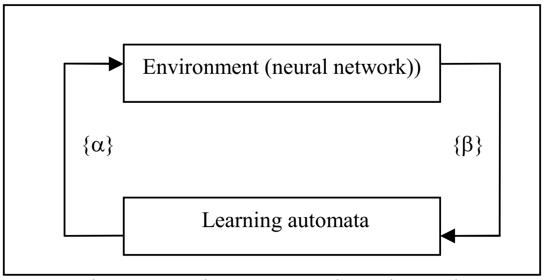

Figure 1. Learning automata and neural network

3. Results and Simulation

For simulation of proposed algorithm we used a computer with processor Intel Pentium(R) CPU, 2.60 GHz, 2.60GHz and RAM was 4 GB and operating system was windows 8. All simulation was performed on MATLAB R2012a 64bit.

In this simulation, sixteen typical fault cases are worked out to contribute the training sample set. For each fault case, the traffic level, sending packets success rates, and broadcasting packets level of source node S; link load level, packets forwarding delay percent and broadcasting packets level of intermediate nodes; link load level, the ratio of handling packets and receiving packets of destination node are taken as the neural network inputs. The states of the nodes in communication (1 source node, 3 intermediate nodes, and 1 destination node) are the outputs. If a certain output approaches to 1, then the corresponding node is considered in fault.

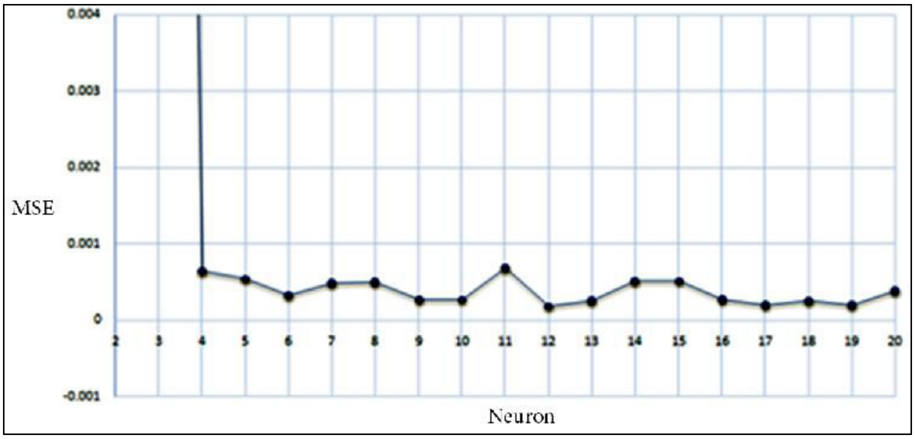

Figure 2- variations of MSE vs number of neurons on hidden layer

3.1. Number of neurons in hidden layer

The number of neurons in hidden layer of a multilayer perceptron influences on output accuracy. So that the number of neurons is less and more than needed has negative impact on network performance. To determine the number of neurons in the proposed method performed simulations for different number of neurons in the hidden layer and the results are shown in Figure 2. As shown 3 neurons in the hidden layer has maximum error (MSE). By increasing the number of neurons in the hidden layer is reduced.

As shown in Figure 2, total of 12 neurons in hidden layer can be appropriate for the proposed network. A result was extracted base on this number of neurons in hidden layer for algorithm proposed in this paper.

3.2. 3-2- Results of standard backpropagation and proposed method

Table 1 shows overall results of standard backpropagation algorithm for 500 times run. Left side of the table, the output of the neural network and the right shows the expected results. The sum of the differences between left and right determine the number of false detection algorithm for backpropagation algorithm, this value is equal to 423.

Table 1. Recognition rate in standard backpropagation method and with learning automata

Output of neural network					Target					LRP-NN				
1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
500	3	4	5	9	500	0	0	0	0	500	5	3	6	1
8	500	7	8	5	0	500	0	0	0	6	491	5	5	2
11	14	500	16	13	0	0	500	0	0	7	4	498	5	3
75	84	68	496	66	0	0	0	500	0	3	6	8	490	2
4	7	5	7	500	0	0	0	0	500	2	6	5	3	496

Table 1 show results of the implementation of the algorithm for 500 times. The advantages of this method is that the number of iterations is less precise, high speed and avoid local optimization of the neural network. Moreover, Table 3 show results of 500 times run of LRP-NN type shows As mentioned, in LRP we have two actions with equal probability at the first, after getting a response automaton (here environment is the modified error of each neuron in the neural network). Using the formulas contained in the automaton will likely change everything. Select an action based on Roulette wheel in the training of the neural networks evolve. The advantage of this method over other Automata is dependent on the number of iterations and converges fast. But lower accuracy than other methods. This condition was shown in figure 3.

3.3. Reliability

Reliability (R) is a time-dependent parameters and thus R(t) is displayed. Defined as a conditional probability that the system works properly as long as the time interval t0 to t, if at time t0 beginning work and works right[17]. Here network run for 500 times and calculated the percentage of right decisions as a below.

$$R\% = \frac{\#diagnosis - \#fault}{\#diagnosis} \times 100 \tag{6}$$

According to above reliability is calculated for each of algorithm and we have:
Standard backpropagation of error detection: 423
No error in LRP -NN: 112

4. Conclusion

Set of mobile nodes in mobile ad hoc networks with wireless receivers and transmitters for communication. Mobile ad hoc networks are known as short-lived networks, mobile networks consist of nodes in the absence of any centralized support. This is a new form of network services in places where it is not possible to make that possible. The final network design fault-tolerant capability is a very important. Articles and much research has been done in this direction. Training neural networks using back propagation algorithm is one of the ways in which it is used. The local search algorithm works very well. The disadvantage of this algorithm is locally optimal exposure. So here are the standard back propagation algorithm using learning automata has been improved. With LA-NN algorithm, fault diagnosis system in MANETs was improved.

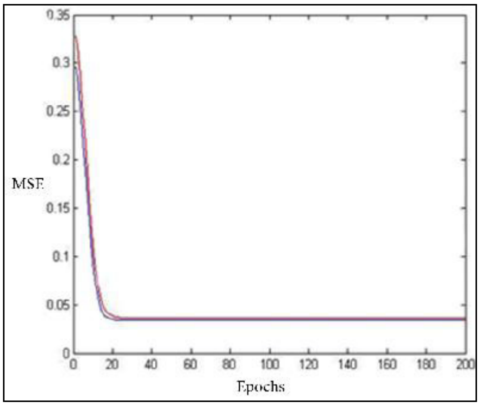

Figure 3. Variations of MSE vs epochs in LRP –NN

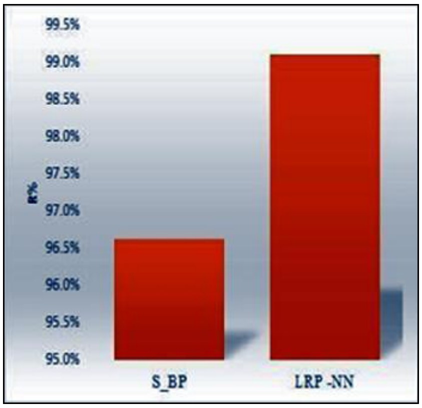

Figure 4. Reliability in proposed methods

References

1. S. Misra, S.K.D, M.S.O, K.V, and U.G, *A low-overhead fault-tolerant routing algorithm for mobile ad hoc networks: A scheme and its* simulation analysis, Simulation Modelling Uractice and Theory, 2010, vol. 18, UU. 637–649.
2. Afsharfarnia, A. and A. Karimi , *A New Clustering Algorithm Using Links' Weight to Decrease Consumed Energy in MANETs*. 2014.
3. AfsharFarnia, A. and A. Karimi , *A Novel Clustering Algorithm for Mobile Ad-Hoc Networks Based on Determination of Virtual Links Weight to Increase Network Stability*, 2014.
4. M. Chouhan, M.N. Sahoo, and U.M. Khilar, *Fault Diagnosis in MANET*, Communications in Computer and Information Science, 2011, vol 190, 119-128
5. A. Vashist, R. Manousakis, K. Chadha, R, Chiang, C.J Serban, C Ali, S.E,*Towards network invariant fault diagnosis in MANETs via statistical modeling: The global strength of local weak decisions*, Presented at the Network Operations and Management Symposium, 2012.
6. F. U. Preparata, G. Metze, and R.T.Chien, *On the Connection Assignment Uroblem of Diagnosable Systems*, Electronic Computers, IEEE Transactions on, vol. EC-16, 1967, pp 848-854.
7. F. Barsi, F. Grandoni, and U. Maestrini, *A Theory of Diagnosability of Digital Systems*, ComUuters, IEEE Transactions on, vol. C-25,1976, pp: 585-593.
8. H. Guangjie, Huang Guangjie, G.Wei Guo Wei, S.Jian Su Jian, *A Novel Fault Diagnosis System for MANET Based on Hybrid GA-BP Algorithm*, 2008, pp, 1-4.
9. B. Mashoufi, M. R. Meybodi, A. Motamedi and M. B. Menhaj, *Neural Networks Engineering UsingLearning Automata: An Adaptive Algorithm for Determination of Optimal Number of Weights inNeural Networks*, Proceedings of Ninth Conferene on Electrical Engineering, Power & WaterInstitute of Technology, 2001, pp. 27-1 to 27-14.
10. M. R. Meybodi, H. Beigy, *Neural Networks Engineering Using Learning Automata: Determination ofOptimal Size for Three Layers Neural Networks*, Proceedings of The Fifth Annual International CSIComputer Conference, Tehran, Iran, 2000, pp. 431-450.
11. M. R. Meybodi, H. Beigy and M. B. Menhaj, *Utilization of Fixed Structure Learning Automata forAdaptation of Learning Rate in Backpropagation Algorithm*, Pakistan Journal of Applied Sciences,Vol. 2, No. 4, 2002, pp. 437-443.
12. M. R. Meybodi, H. Beigy, *New Algorithms for Adaptation of Backpropagation AlgorithmParameters*, Iranian Journal of Science& Technology, Transaction B, vol.25, No. B3,2001, pp. 515-532.
13. B. Mashoufi, M. B. Mehaj, A. Motamedi and M. R. Meybodi, *Introducing A Novel Learning Automaton Based Method for Adapting VLR Learning Algorithm Parameters for MLP NeuralNetworks*, Jounal of Amirkabir,2002, Vol. 13, No. 51, pp. 398-412.
14. H.Beigy, M.R. Meybodi, M. B. Menhaj, *Utilization of Fixed Structure Learning Automata forAdaptation of Learning Rate in Backpropagation Algorithm*, Proc.of International NAISO Symp. OnInformation Science Innovations in Engineering & Natural and Artificial Intelligent Systems(ENAIS'2001), Dubai, U.A.E., 2001, pp. 661-667.
15. H. Beigy and M. R. Meybodi, *Experimentation on Learning Automata Based Methods forAdaptation of BP Parameters*, Proceedings of Ninth Conference on Electrical Engineering, Power &Water Institute of Technology, 2001, pp. 4-1 to 4-9. 8-10.
16. H. Beigy,M. R. Meybodi, *Adaptation of Parameters of BP Algorithm Using Learning Automata*,Proceedings of the 7th Joint Symposium on Neural Computation, Institute for Neural Computation,University of California, San Diego, 2000, p. 4.
17. Karimi, A. and F. Zarafshan , Accurate *and efficient reliability Markov model analysis of predictive hybrid m-out-of-n systems*, Computer Science and Information Technology (ICCSIT), 3rd IEEE International Conference on, IEEE, 2010.