International Journal of Advanced Research in Engineering Technology & Science

ISSN: 2394-2819

Email: editor@ijarets.or Volume-4, Issue-12 December- 2017 www.ijarets.org

SP ROUTING ALGORITHM BASED ON ARTIFICIAL NEURAL NETWORK

MANIK CHANDRA PANDEY[1], NISHA[2], AMIT SHARMA[3]

Department of Computer Science & Engineering

IIMT College of Engineering,

Greater Noida

ABSTRACT - Shortest path problem is the optimization problem arising in numerous planning and designing contexts. This paper presents artificial neural networks based routing for solving the shortest path problem. The recurrent neural network is able to generate solutions for the shortest path problem. The performance of the neural network is established by examples. In this technique Firstly we calculate the weights of unweighted graph using artificial neural network algorithm. After that we apply Dijkastra's algorithm to find the shortest path routing.

Keywords*:* Shortest path, Neural network, Optimization, Learning rates, Routing algorithms, Dijkastra's

INTRODUCTION

In modern communication networks, particularly in the packet switched networks, routing is important process that has a significant impact on the network's performance. Many routing algorithm comprises finding the "optimal" path(s) between sources to destination router, enabling high-speed data transmission to avoiding a packet loss. Problem of finding the shortest path between the nodes is a well-known problem in network analysis. Shortest path optimization is one of the most significant issues that have a major impact on the network's act. An ideal routing algorithm should attempt to find an optimal path for packet transmission within a particular time to satisfy the Quality of Service (QoS). Routing is a method of paths discovery between nodes in network. Generally there are mainly two types of policies in routing - static and dynamic routing. In static routing, the paths between the nodes are pre computed based on fixed factors for example number of nodes, buffer space, bandwidth etc. and are stored in routing table. Every packet between any two nodes follows the same path. Static routing fails when network topology changes hence the path between two nodes may also change. But in dynamic routing policy, the routes are generated when required but not stored. The generation of new routes are based on the factors like traffic, link utilization, jitter, bandwidth, delay etc which is meant at having highest performance. For message transmission routing policy may be centralized or distributed. In the case of centralized routing, generation of routes between any pair of nodes done by only centralized node. . Centralized routing is not sufficient in IP networks because before route computation it is required to collect whole network state, which is very tuff task but In distributed routing, generation of routes is independent by each node between pair of nodes as per requirement. Other classification of routing policy is optimal routing (global routing) and shortest path routing (local routing). Some other shortest path algorithms are distance vector algorithm and link state algorithm. Each node in the network follows the policy of store and forward. The link recital can be measured in terms of link delay or bandwidth. The topology of the network may change due to growth in number of nodes, or failure of node. This change in topology should be reflected in the routing table, which helps the routing protocol to generate optimal route for the current state of network.

PROBLEM STATEMENT

Distance between the nodes are being calculated with the concept of Rosenblatt algorithm, then calculated weights will be used as the distance between the nodes and Dijkastra's algorithm will be applied to calculate the shortest distance between the nodes.

PROPOSED ALGORITHM

Human brain is getting experienced to adapt themselves to their surrounding environments. So as a result the information processing capability of the brain is rendered, when this happen the brain becomes plastic.

1. Plastic: - Capability to process information, capability of adding. Must preserve the information it has learn previously.
2. Stable: - Remain stable when it is presented with irrelevant information, useless information.

Synapses with large area are excitatory (+Ve weights) & with small area are inhibitory (-Ve weights). Synapses of the neuron are modulated as weights. (Strength of the connection) Biological neuron receives all inputs through dendrites, sums them & produces an output. If the sum is greater than the threshold value, then input signals are passed to the cell body. NN can map input patterns into output patterns. NN's are robust systems. They can recall full patterns from incomplete pattern or noise channel.

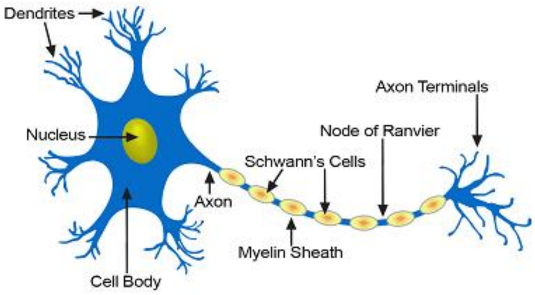


Fig1: Biological neural network

Perceptron is the generic name given by the Frank Rosenblatt because it is a model of EYE. Perceptron was an attempt to understand human memory, learning & cognitive process. Training algorithms of perceptron is supervised learning.

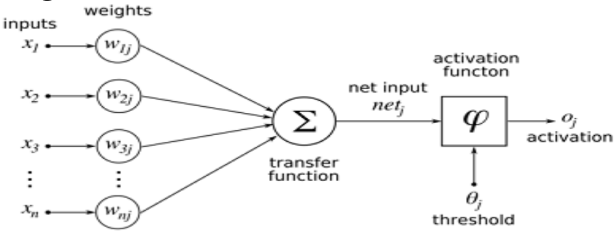


Fig2: ANN with Activation Function

It accept no. of inputs Xi (i=1, 2, 3…………..n) & compute a weighted sum of these inputs. The sum is then compared with a threshold θ of an output Y (which is either 0 or 1)

$$Y = 1 \quad \text{if} \quad \sum^{n} Wi*Xi \geq \theta \qquad i=1$$

$$Y = 0 \quad \text{if} \quad \sum^{n} Wi*Xi \leq \theta \qquad i=1$$

The perceptron is the single transmission networks consist of sensor unit, association unit & output of response unit.

The sensor unit produce a binary output 0 or 1. Association unit behave like a basic building block. Response unit comprise pattern recognition. Thus the O/P of the response unit could be such that is the weighted sum of the input is less the n or equal to 0, then the output is 0 else 1.

Proposed Algorithm:

Step 1: create perceptron with (n+1) input neurons X0 X1 …Xn where X = 1 is the bias input. Let O be the output neuron.

Step 2: Initialize W= (Wo, W1…, Wn) to random weights.

Step 3: Iterate through the input patterns X of the training set using the weight set (i.e.) compute the weighted sum of input net j =Sum(XiWi) for each input pattern j.

Step 4: Compute the output Y using the step function

Y= f (net j) = 1 net j > 0
=0 otherwise

International Journal of Advanced Research in Engineering Technology and Sciences ISSN 2349-2819
www.ijarets.org Volume-4, Issue-12 December- 2017 Email- editor@ijarets.org

Step 5: compare the computed output Yj with the target output Yj for each input pattern j. If all the input patterns have been classified correctly output the weights and exist.
Step 6: Otherwise, update the weights as given below:
if the computed output Yj is 1 but should have been 0 ,
Wi= Wi - α.xi
if the computed output Yj is 0 but should have been 1 ,
Wi= Wi + α.xi, where α is the learning rate.
Step 7: Goto step 3.

DIJIKASTRA ROUTING

Algorithm assigns to every node j a pair of labels (pj, dj) in which pj is the node preceding node j in the existing shortest path from 1 to j, dj is the length of this shortest path. Some of the labels called temporary, i.e. they could change at a future step; some labels are called permanent, i.e. they are fixed and the shortest path from 1 to a node that is permanently labeled has been found.

Step 1 Label node 1 with the permanent labels (□, 0). Label every node j, such that (1,j) is an arc in the graph, with temporary labels (1, d1j). Label all other nodes in the graph with temporary labels.

Step 2 Let j is a temporarily labeled node with the minimum label dj, i.e. dj=min {dl: node l is temporarily labeled}. For every node k, such that (j,k) is in the graph, if dk> dj+djk then relabeling k as follows:
pk=j, dk=dj+djk.
Consider the labels of node j to be permanent.

Step 3 Repeat step 2 until all nodes in the graph are permanently labeled.

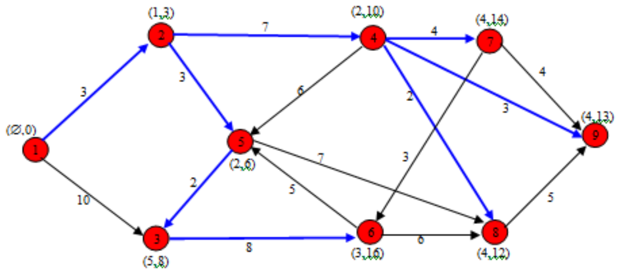


Fig3: Dijikstra algorithm routing

The shortest paths can be found by reading labels pj.
Example Find the shortest paths from node 1 to all other nodes.

LEARNING RATE AND METHOD

Following are the learning methods: -

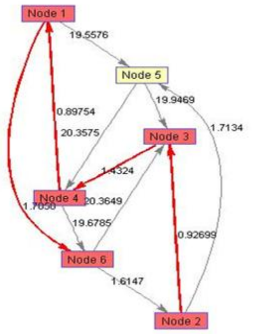
1. Supervised learning: - Every input pattern is associated with an output pattern. A teacher is present during the learning process. A comparison is made between computed & desired output. Error can be used to change the network parameter so performance is improved.
2. Unsupervised learning: -There is no target output NN systems learn by its own by discovering the feature in the input patterns.
3. Reinforcement: - There is a teacher but no desired outputs. The teacher only indicates that the computed output is correct or not. Reward for the correct answer & penalty for the wrong answer.
4. Supervised Gradient descent: - Error E is minimized in terms of the weights of the activation function. If Δ Wij = η ∂ E/∂ Wij.

Most of the network structure undergoes learning procedure during which synaptic weights W and v are adjusted. Learning rate coefficient determines the size of the weights adjustments made at each iteration and hence influences the rate of convergence. Poor choice of coefficient can result in a failure in convergence. If learning rate coefficient too large the search path will oscillate and convergence more slowly in a direct descent. If the coefficient is too small the descent will progress in small steps significantly increasing time to converge.

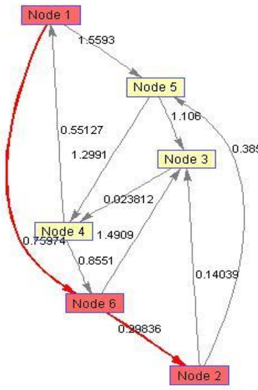
RESULT

(1) DATA SET FOR 6 NODES AND 11 EDGES

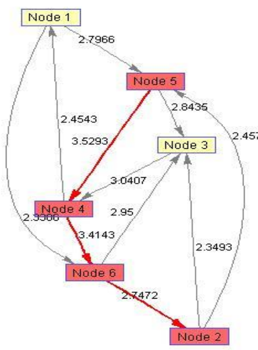
I/O SET FOR ROSENBELATT ALGORITHM					I/O SET FOR DIJIKASTRA ALGO		
W_0	N		X_2	W_1, W_2	SN, DN	DIST	SPN
2	0.7	4	3 2	0.8, 19.4	2, 6	4.96	23416



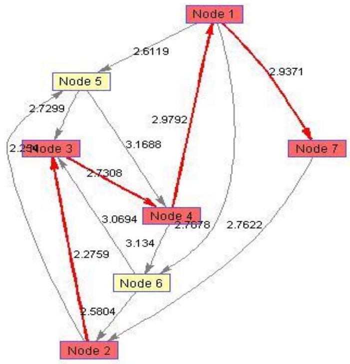
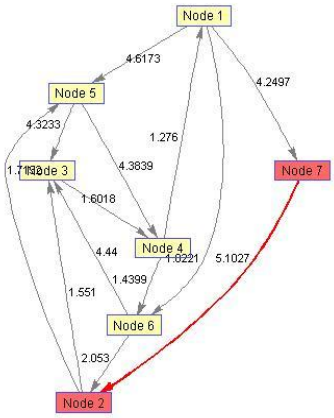
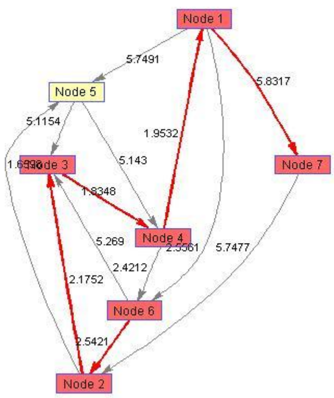
2	0.6	3	6	-0.2,0.6	1,2	1.05	162



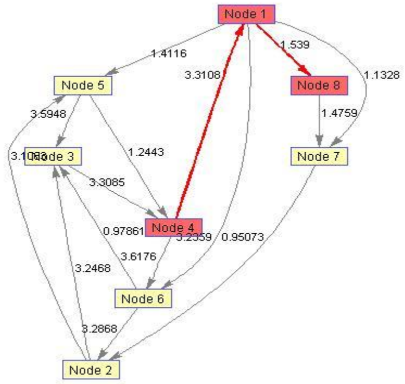
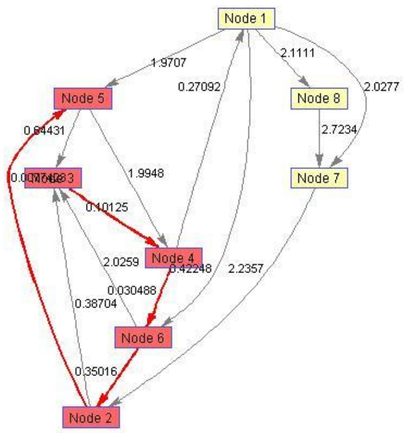
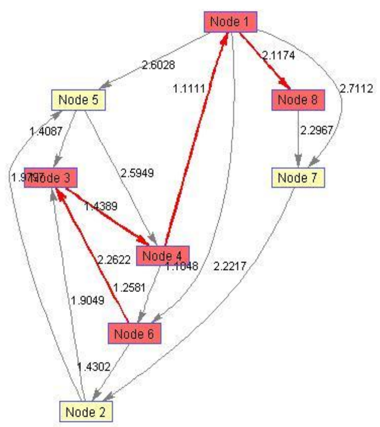
3	0.7	3	4	2.2,2.6	5,2	9.69	5462



(2) DATA SET FOR 7 NODES AND 13 EDGES

<table>
<tr><td colspan="5">I/O SET FOR ROSENBELATT ALGORITHM</td><td colspan="3">I/O SET FOR DIJIKASTRA ALGO</td></tr>
<tr><td>W_0</td><td>n</td><td></td><td>X_2</td><td>W_1, W_2</td><td>SN, DN</td><td>DIST</td><td>SPN</td></tr>
<tr><td>3</td><td>0.7</td><td>3</td><td>4</td><td>2.2, 2.6</td><td>2, 7</td><td>10.9</td><td>23417</td></tr>
<tr><td colspan="8">
</td></tr>
<tr><td>4</td><td>0.8</td><td>4</td><td>9</td><td>1.2,4.2</td><td>7,2</td><td>5.1</td><td>72</td></tr>
<tr><td colspan="8">
</td></tr>
<tr><td>5</td><td>0.9</td><td>4</td><td>9</td><td>1.6,5.1</td><td>6,7</td><td>14.3</td><td>623417</td></tr>
<tr><td colspan="8">
</td></tr>
</table>

(3) DATA SET FOR 8 NODES AND 15 EDGES

<table>
<tr><td colspan="5">I/O SET FOR ROSENBELATT ALGORITHM</td><td colspan="3">I/O SET FOR DIJIKASTRA ALGO</td></tr>
<tr><td>W_0</td><td>n</td><td></td><td>X_2</td><td>W_1, W_2</td><td>SN, DN</td><td>DIST</td><td>SPN</td></tr>
<tr><td>4</td><td>0.1</td><td>24</td><td>18</td><td>2.6, 0.6</td><td>4, 8</td><td>4.8</td><td>418</td></tr>
<tr><td colspan="8">
</td></tr>
<tr><td>6</td><td>0.2</td><td>3</td><td>8</td><td>-0.2,1.8</td><td>3,5</td><td>.48</td><td>34625</td></tr>
<tr><td colspan="8">
</td></tr>
<tr><td>1</td><td>0.5</td><td>6</td><td>10</td><td>1.0,2.0</td><td>6,8</td><td>6.92</td><td>63418</td></tr>
<tr><td colspan="8">
</td></tr>
</table>

CONCLUSION

In this article, we have proposed a new method for solving the shortest path problem in computer networks. The approach based on the concept of Rosenblatt model with Dijikstra. The algorithm uses a modified indirect path-encoding scheme. This algorithm is simple and easy and can find quickly the shortest path without falling into local minimums, which occur if we use energy function to solve the same problem. It is very easy to run the new algorithm on parallel computer or even on neural computer. That will also reduce the computational time drastically.

FUTURE WORK

Thus with the invention of such tactic the shortest path can be computed between two nodes even if the weights are unknown. In the future we can apply neural network to find out the all pair shortest path to find out the shortest path in a given graph where the weights are unknown.

ACKNOWLEDGMENT

Author would like to thank Prof. Deepak Tomer, Dept. of Computer Science & Engineering, TECHNOCRATS INSTITUTE OF ENGINEERING & TECHNOLOGY, BHOPAL, MADHYA PRADESH, INDIA for his support. I feel grateful for the support of Prof. Deepak Tomer for writing and publishing paper. The author will also like to express sincere appreciation and gratitude to Dept. of Computer Science & Engineering, TIT, Bhopal, M.P. India, which has provided tremendous assistance throughout the work.

REFERENCES

1. D. Kim, J. Garcia and K. Obraczka, “Routing Mechanisms for Mobile Ad Hoc Networks based on the Energy Drain Rate”, IEEE Transactions on Mobile Computing. Vol 2, no 2, 2003, pp.161-173.
2. Ad Hoc Mobile Wireless Networks: Protocols and Systems By**:** C.-K. Toh **-** Ph.D. ,**Publisher:** Prentice Hall
3. L. Bodin, B. L. Golden, A. Assad, and M. Ball, “Routing and scheduling of vehicles and crews: The state of the art,” *Computer. Res.*, Vol. 10, pp. 63–211, 1983.
4. A. Ephremides and S. Verdu, “Control and optimization methods in communication network problems,” *IEEE Trans. Automat. Contr.*, Vol. 34, pp. 930–942, 1989.
5. J. K. Antonio, G. M. Huang, and W. K. Tsai, “A fast distributed shortest path algorithm for a class of hierarchically clustered data networks,” *IEEE Trans. Computer.*, Vol. 41, pp. 710–724, 1992.
6. S. Jun and K. G. Shin, “Shortest path planning in distributed workspace using dominance relation,” *IEEE Trans. Robot. Automat*, Vol. 7, pp. 342–350, 1991.
7. P. Soueres and J.P. Laumond, “Shortest paths synthesis for a car-like robot,” *IEEE Trans. Automat. Contr.*, Vol. 41, pp. 672–688, 1996.
8. Pandey A.,Bansal K.K.(2014): “Performance Evaluation of TORA Protocol Using Random Waypoint Mobility Model” *International Journal of Education and Science Research Review* Vol.1(2)
9. Tiwari S.P.,Kumar S.,Bansal K.K.(2014): “A Survey of Metaheuristic Algorithms for Travelling Salesman Problem ” *International Journal Of Engineering Research & Management Technology* Vol.1(5)
10. Yadav R.K.,Bansal K.K.(2012) : “Analysis of Sliding Window Protocol for Connected Node” *International Journal Of Soft Computing and Engineering(IJSCE)* Vol.2(5) pp.292-294
11. W. Stalling, *High-Speed Networks: TCP/IP and ATM Design Principles.* Englewood Cliffs, NJ: Prentice-Hall, 1998.
12. M. K. Ali and F. Kamoun, "Neural networks for shortest path computation and routing in computer networks," *IEEE Trans. Neural Networks,* Vol. 4, pp. 941-954, Nov. 1993.
13. D. C. Park and S. E. Choi, "A neural network based multi-destination routing algorithm for communication network," in *Proc. Joint Conf. Neural Networks,* 1998, pp. 1673-1678.
14. C. W. Ahn, R. S. Ramakrishna, C. G. Rang, and I. C. Choi, "Shortest path routing algorithm using hopfield neural network," *Electron. Lett.*, Vol. 37, no. 19, pp. 1176-1178, Sept. 2001.
15. M. Munemoto, Y. Takai, and Y. Sato, "A migration scheme for the genetic adaptive routing algorithm," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics,* 1998, pp. 2774-2779.
16. Y. Leung, G. Li, and Z. B. Xu, "A genetic algorithm for the multiple destination routing problems," *IEEE Trans. Evol. Comput,* Vol. 2, pp. 150-161, Nov. 1998.
17. Rolf Pfeifer, Dana Damian, Rudolf Fuchslin, Neural Network.
18. “Exploring Genetic Algorithm for Shortest Path Optimization in Data Networks” Global journal of computer science and technology volume 10 issue 11 ver(1.0)October 2010 by Dr. Rakesh Kumar, Mahesh Kumar.
19. Park, Y.-K., & Lee, G. (1995). Applications of neural networks in high speed communication networks. IEEE Communications Magazine, 33(10), 68–74.
20. Marbach P., Mihatsch O, & Tsitsiklis, J. N. (2000). Call admission control and routing in integrated service networks using neuro-dynamic programming. IEEE Journal of Selected Areas of Communication