Study of Path Optimization in Packet Switching Network using Neural Network

Zaiba Ishrat
Department of Electronics & Communication, RGGI, Meerut U.P., India

Sarvesh Kumar Sharma
Department of Electronics & Communication, RGGI, Meerut U.P., India

ABSTRACT

The problem of finding the optimal path between two nodes is a well known problem in network analysis. Optimal routing has been widely studied for interconnection networks this paper work considers the problem of finding the optimal path. A Genetic algorithm based strategy is proposed and the algorithm has been developed to find the Optimal Path. This paper work presents a neural network based approach to the shortest path routing problem. Weights adjustment of the neurons has been used for solving the problem of optimum path. This paper presents the back propagation algorithm to solve the problem of optimum path in multi layer feed forward (MLFF) network. Even though shortest path routing algorithms are already well established, there are researchers who are trying to find alternative methods to find shortest paths through a network. One such alternative is to use of neural network.

Keywords: shortest path, neural network, optimization, packet switching, mlff, activation function, learning rates, algorithms.

1. INTRODUCTION

In modern communication networks, particularly in packet switched networks, routing is an important process that has a significant impact on the network's performance.Ideal routing algorithm comprises finding the "optimal" path(s) between source and destination router, enabling high-speed data transmission and avoiding a packet loss. The problem of finding the shortest path between two nodes is a well-known problem in network analysis. Shortest path algorithms have been a subject of extensive research, resulting in a number of algorithms for various conditions and constrain [1-3]. In a packet switching network, communication between two hosts generally takes place in the following manner: the transmitting host delivers to a node a block of data, called a packet, which are addresses to the destination host. The objective of a routing strategy is essentially to minimize the mean delay of the packets in a network, subject to some reliability or capacity constraints [3-4]. Routing is one of the most important issues that have a significant impact on the network's performance [5], [6]. An ideal routing algorithm should strive to find optimum path for packet transmission within a specified time so as to satisfy the quality of Service (QoS) [8]-[9]. There are several search algorithms for the shortest path (SP) problem: the O(n2) Bellman's dynamic programming algorithm for directed a cycle networks, the O(n2) Dijkstra-like labeling algorithm and the O(n3) Bellman–Ford successive approximation algorithm for networks with nonnegative cost coefficients only, where n denotes the number of vertices in the network. In most of the current packet-switching networks, some form of SP computation is employed by routing algorithms in the network layer [8], [9]. Specifically, the network links are weighted, the weights reflecting the link transmission capacity, the congestion of networks and the estimated transmission status such as the queuing delay of head-of-line (HOL) packet or the link failure. The SP problem can be formulated as one of finding a minimal cost path that contains the designated source and destination nodes. In other words, the SP routing problem involves a classical combinatorial optimization problem arising in many designs and planning contexts [10]-[11]. Since neural networks (NNs) [10]-[11] promise solutions to such complicated problems.

2. MULTILAYER FEED FORWARD NETWORK

The network, as its name indicates made up of multiple layers. Thus architecture of this class besides processing an input and an output layer also have a hidden layer known as hidden neurons or hidden units. The input layer neurons are linked to the hidden layer neurons and weights on these links are referred to as input –hidden layer weights. Again the hidden layer neurons are connected to the output layer neurons and corresponding weights referred as hidden output layer weight [15]. Fig [1]

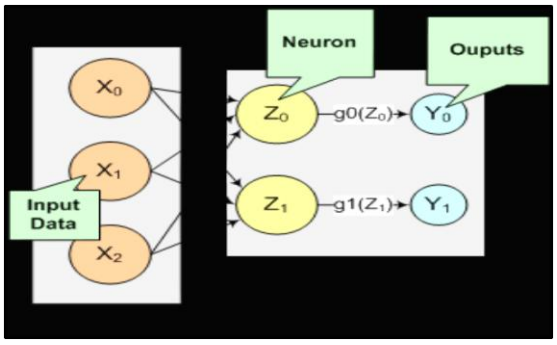

Fig [1] Multilayer feedforward network

3. BACKPROPAGATION NETWORK

Backpropagation is the systematic method of training the multilayer artificial neural network. It is built on high mathematical foundation and has very good application potential.

Analogous to a biological neuron, artificial neurons receives much input representing the output of the other neurons. Each input is multiplied by the corresponding weights analogous to the synaptic strengths. All of this, weighted input are than summed up and passed through an activation function to determine neurons output fig[2].

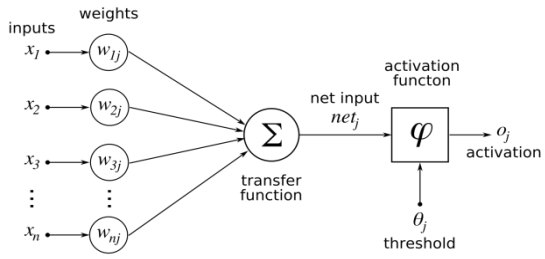

Fig [2]

U(t)= W $_1$ i$_1$ +W $_2$ I $_2$+……..+W $_n$ I $_n$

U=[W][I]

considering the threshold, the relative input to the neurons is given by

U(t)= W $_1$ i$_1$ +W $_2$ I $_2$+……..+W $_n$ I $_n$ - Θ

$$= \sum_{i=0}^{n} WiIi$$ where Wo= - Θ; Io=1

The output using non linear transfer function f is given by

O=f(u)

4. ACTIVATION FUNCTION

To generate the final output the sum is passed through the non linear filter called the activation function or transfer function which releases the output. A common choice is the sigmoidal or logistic function,[15]

$$Ø(i)=\frac{1}{1+e^{-ai}}$$

Where α is the slope parameter, which adjusts the abruptness of function as it changes between the two asymptotic values.fig[3]

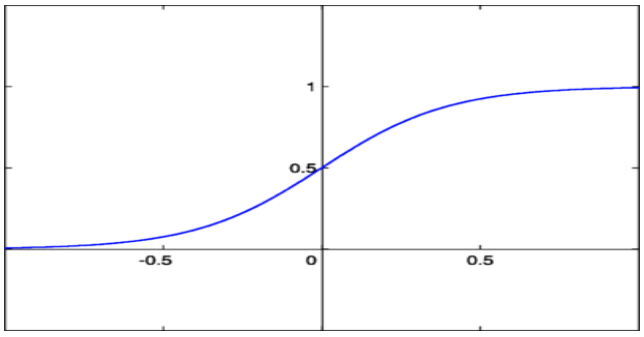

Fig [3] [logistic function]

One more activation function is step function or Heaviside function and is such that fig[4]

Ø(i)= 1, I>0

= 0, I≤0

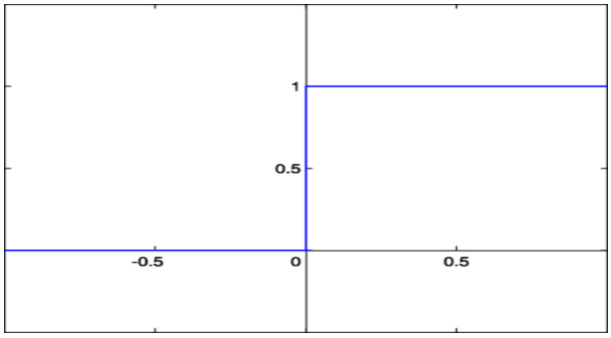

Fig [4] [Heaviside function]

5. LEARNING RATE

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.

6. PROBLEM STATEMENT

Consider a weighted direct graph G = (V; E) where V is a set of n vertices and E is an ordered set of m edges. A fixed cost cij is associated with the edge from vertices i to j in the graph G. In a transportation or a robotic system, for example, the physical of the cost can be the distance between the vertices, the time or energy needed for travel from one vertex to another. In a telecommunication system, the cost can be determined according to the transmission time and the link capacity from one vertex to another. In general, the cost coefficients matrix [cij] is not necessarily symmetric, i.e., the cost from vertices i to j may not be equal to the cost from vertices j to i. Furthermore, the edges between some vertices may not exist, i.e., m may be less than n2 (i.e., m < n2). The values of cost coefficients for the nonexistent edges are defined as infinity. More generally, a cost coefficient can be either positive or negative. A positive cost coefficient represents a loss, whereas a negative one represents a gain. It is admittedly more difficult to determine the shortest path for a network with mixed positive and negative cost coefficient [7].

Learning algorithm:

If the output is correct then no adjustment of weights is done.

$$\mathbf{W_{ij}^{(k+1)} = W_{ij}^{k}}$$

If the output is 1 but should have been 0 then the weights are decreased on the active input links.

$$\mathbf{W_{ij}^{(k+1)} = W_{ij}^{k} - \alpha.x_i}$$

Where α is the learning rate.

If the output is 0 but should have been 1 then the weights are increased on the active input links.

$$\mathbf{W_{ij}^{(k+1)} = W_{ij}^{k} + \alpha.x_i}$$

$W_{ij}^{(k+1)}$ is new adjusted weight and W_{ij}^{k} is old weight.

Rosenblatts algorithm:

Step 1: create perceptron with (n+1) input neurons X_0 X_1 …X_n where X = 1 is the bias input. Let O be the output neuron.

Step 2: Initialize W= (W_o,W_1…W_n) 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

$net_j = \sum_{i=0}^{n} XiWi$ 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

Step 5: compare the computed output Y_j with the target output Y_j 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 Y_j is 1 but should have been 0 ,

$Wi=Wi - \alpha.x_i$

if the computed output Y_j is 0 but should have been 1 ,

$Wi=Wi + \alpha.x_i$

where α is the learning rate.

Step 7: Goto step 3.

7. CONCLUSION

The neural network has promised an easy way to optimize the packet switch multilayer network. A preliminary implementation of the Rosenblatts algorithm lays a foundation for further optimization in SP using neural network. The network learns optimized route itself after few iterations.

8. REFERENCES

[1] E.W. Dijkstra, “A note on two papers in connection with graphs,” Numeriske Mathematics 1 pp.269-271 1959.

[2] D.Eppstein, “Finding the k shortest paths,” SIAM journal on Computing 28(2) pp.653-674 1998.

[3] R.W Floyd, “Algorithm97: Shortest paths, “Communications of the ACM 5 pp.345-357 1962.

[4] Baransel C,Dobosiewicz W, Gburzynski p, “Routing in multihop packet switching networks:Gb/s challenges,”IEEE Network 9(3) pp.38-61 1995.

[5] Suk-Gwon C, “Fair integration of routing and flow control in communication networks,”IEEE Trans Commun 40(4) pp. 821-34 1992.

[6] K. B. Kumar and J. Jaffe, “Routing to multiple destinations in computer networks”, IEEE Transactions on Communications, vol. 31,no. 3, pp. 343-351, 1983.

[7] 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.

[8] 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.

[9] C. W. Ahn, R. S. Ramakrishna, C. G. Rang, and I. C. Choi, "Shortest path routing algorithm using hopfield neural neural network," Electron. Lett., vol. 37, no. 19, pp. 1176-1178, Sept. 2001.

[10] 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.

[11] J. Inagaki, M. Haseyama, and H. Kitajima, "A genetic algorithm for de-termining multiple routes and its applications," in Proc. IEEE Int. Symp. Circuits and Systems, 1999, pp. 137-140.

[12] 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.

[13] G. Syswerda, "Uniform crossover in genetic algorithms," in Proc. 3rd Int. Conf. Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1989, pp. 2-9.

[14] M. Parsa and Q. Zhu, “An iterative algorithm for delay-constrained minimum-cost multicasting”, IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 461 – 474, 1998.

[15] Neural Networks,Fuzzy logic and Ganetic algorithm by S.Rajasekaran and G.A. Vijjylakashmi pai H. Rauch and T.Winarske, “Neural networks for routing communication traffic,” IEEE Cont. Syst. Mag., pp. 26–30, Apr. 1988.

[16] P. Soueres and J.-P. Laumond, “Shortest paths synthesis for a car-like robot,” IEEE Trans. Automat. Contr., vol. 41, pp. 672–688, 1996.

[17] E. L. Lawler, Combinatorial Optimization: Networks and Matroids. New York: Holt, Rinehart, and Winston, 1976, pp. 59–108.

[18] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization problems,” Biol. Cybern., vol. 52, pp. 141–152, 1985.