Volume 3, Issue 5, May 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Congestion Control in Wired Network for Heterogeneous resources using Neural Network

Saxena Sachin Kumar*
Department of IT,
COER, Uttarakhand, India

Kumar Dhaneshwar**
Department of IT,
COER, Uttarakhand, India

Krishna Garima#,
Department of CSE,
COER, Uttarakhand, India

Goel Neha$
Department of IT,
COER, Uttarakhand, India

Sati Ayush@
Department of IT,
COER, Uttarakhand, India

Abstract--The wireless network is most imperative parameter in communication among the computers. The use of internet services for time sensitive applications like voice and video requires the quality of service. The TCP/IP also helped in providing the services structure achieve this target. However, network congestion control is limited and comes from the high priority. Some literatures are still looking for replacement techniques such as random early detection (RED) and its modification to manage congestion. In this paper we present neural network control research results to implement on real world. We found that with neural network we can perform better for discrimination acts to cancel the packets for gathering traffic flow, and also provide better quality services to all types' different traffic. This paper presents an extensive analysis of congestion controls recently proposed in literature. To reach our goal, we have implemented one representative algorithm among the so-called TCP-friendly congestion controls. We have then evaluated their TCP-fairness and validation in a Wireless Local Area Network (WLAN).

Keywords: Time Series Neural Network, Scheduling algorithm, releasing time and processing time.

1. Introduction

In order to control the congestion among the wireless network one is needed very complex calculations. The dynamic neural network algorithm presented in this paper is trained to regulate the flow of data close to a reference value determined by network requirements [6]. After training, the neural network operates as an adaptive controller under changes in TCP dynamics. We choose a multi-layer dynamic neural model because of its well-known advantages. This model has been popular since the mid 1990's in many applications for dynamical time-varying and nonlinear systems.

2. Methodology

The methodology used in this paper is neural network designed which must be trained to optimize a TCP network performance measure and other wireless parameters [5]. During network training, the weights and the bias are iteratively updated until they reach their optimal values.

2.1. DEFINE NEURAL NETWORK

Neural Network provides tools for designing, implementing, visualizing, and simulating neural networks. Neural networks are used for applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control [4]. The Neural Network supports feedforward networks, radial basis networks, dynamic networks, self-organizing maps, and other proven network paradigms. In the learning algorithm for the proposed network and the rules for updating the network weights and bias are congestion control parameters such as BER (Bit error Rate), RTT (Round Trip Time) are used. There are mainly two methods for training neural networks:

- a back-propagation-trough-time algorithm
- feed forward learning algorithm
- Learning process

2.2. NEURAL NETWORK ARCHITECTURE

Neural network is basically used to design the attributes of wireless network and simulations are used for clustering, and data-fitting tools [3]. Supervised networks including feedforward, radial basis are deployed to map the network. The architecture of neural network includes input layers, hidden layers and output layers as depicted from fig 1 we have several outputs for various results.

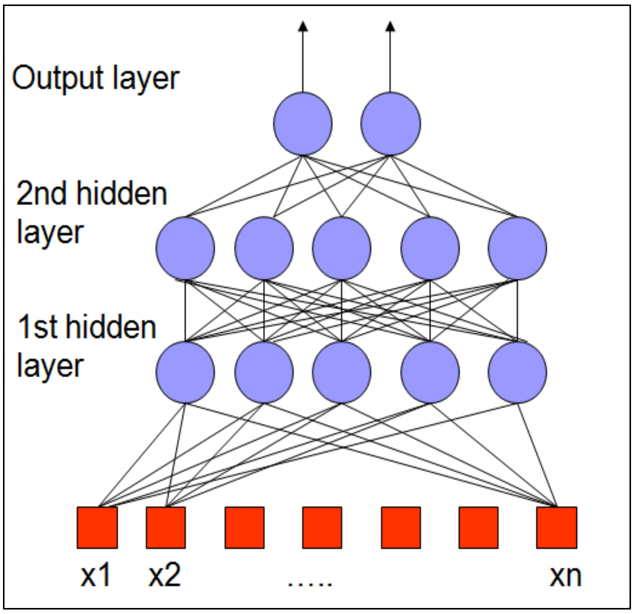

Figure 1. Architecture of Neural Network

The activation function used of these feed forward neural networks is a sigmoid function as in equation:-

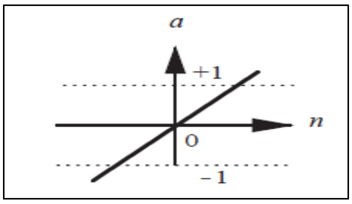

Figure 2. a=purelin(n)

2.3. TRAINING

The neural network designed in Section II must be trained to optimize a TCP network performance measure [2]. During the network training, the weights and the bias are iteratively updated until they reach their optimal values. In this section, learning method for the proposed network and derive the rules for updating the network weights and bias.

2.4. TESTING

To test the data we have certain Matlab commands along with Neural Network Toolbox which are given below:-
Consider this set of data:
p = [-1 -1 2 2; 0 5 0 5]
t = [-1 -1 1 1]
where p is input vector and t is target.

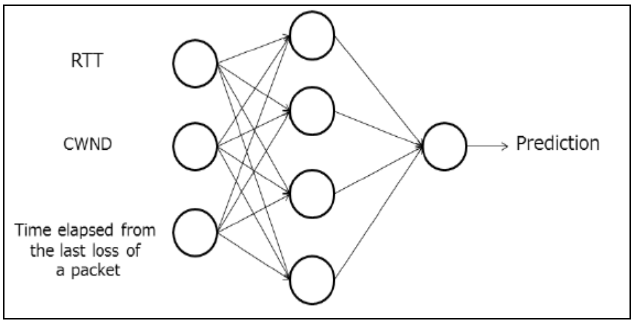

Figure 3. Routes of packets

3. Application Model

In a communication network information is transferred from one node to another as data packets. Packet routing is a process of sending a packet from its source node (s) to its destination node (d). On its way, the packet spends some time waiting in the queues of intermediate nodes while they are busy processing the packets that came earlier. Thus the delivery time of the packet, defined as the time it takes for the packet to reach its destination, depends mainly on the total time it has to spend in the queues of the intermediate nodes [6]. Normally, there are multiple routes that packet could take, which means that the choice of the route is crucial to the delivery time of the packet for any (source, destination) pair. Routing algorithms are methods for finding the best way from a node s to another node d. This may be via a large number of other nodes or it may be in the next sub network. On a small, simple network the problem is almost trivial, statically allocating routes and defining them by hand, but when dealing with a huge internetwork such as the Internet this is not possible. Calculating the best route through such a complex system is computationally difficult and impossible to do by hand, if part of network becomes over filled with packets it can become impossible for packets to move. The queues into which they would be accepted are always full. This is called congestion.

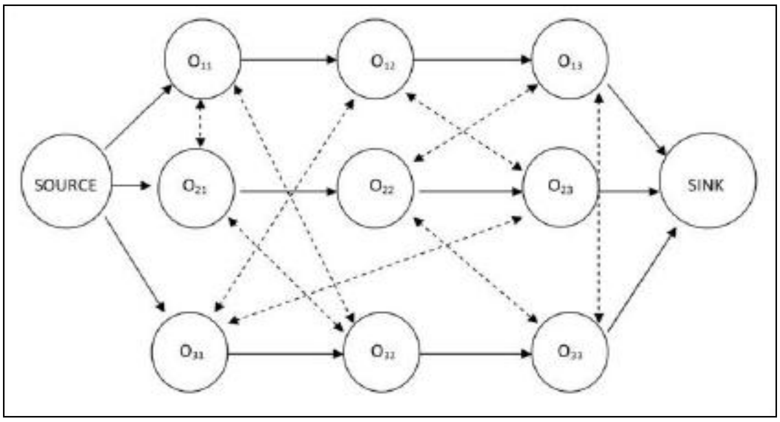

Figure 4. Packets travelling from source to sink

One of the major problems for most communications networks lies in defining an efficient packet routing policy. Routing policy should be able to take into account the congestion. It sends the packet through route that may be long in terms of hops but results in shorter delivery time [7]. The aim of the good routing algorithm is to minimize the effect of network congestion. But, the main problem of conventional routing algorithms that if costs are assigned in a dynamic way, based on statistical measures of the link congestion state, a strong feedback effect is introduced between the routing policies and the traffic patterns. This can lead to undesirable oscillations [10]. The neural networks have available structure of implementing a learning mechanism for data communication networks. For that, a neural network can achieve great accuracy in predicting one particular network problem, namely congestion [1]. Considerable research has been devoted toward solving the routing problem and to manage the congestion in the computer networks. propose using learning techniques to predict congestion problems in computer network [2]. They suggest a simple feedforward neural network to predict congestion in a network a routing & congestion control scheme is proposed for wireless multihop ad hoc network. Neural networks have also been used to decide if certain requests will satisfy the Quality of Service (QoS) parameters set by the network administrators. QoS usually involves classifying network traffic and setting certain restrictions on the traffic based on its class. The predictive powers of neural networks are used to predict if QoS standards will be upheld with the entry of new traffic. The authors of do not address predicting congestion causes, nor do they need to make any attempt towards correction [3]. Their solution falls in the category of open loop solutions, those that prevent problems by not allowing the network to enter into any state in which a transition into a problem state is possible. Limiting network activities to states which could only transition to positive states can vastly reduce the utilization of the network, given the unpredictability of networking traffic. The simulated network is arranged such that six sending nodes are connected to one receiving node through several links which direct the packets to the destination.

4. Results And Discussion

In this work, we set out to show that a neural network is a viable method of implementing a learning mechanism for data communication networks. We have illustrated, through the use of a network simulator, that a neural network can achieve great accuracy in predicting one particular network problem, namely congestion [4]. We realize many more problems exist that for which this approach is applicable, but predicting congestion is just the first step towards our research goals. We also have shown one situation in which a carefully constructed neural network can achieve above average results when structural information about the actual data network is used to form.

Table 2. Actual and Predicted values comparison

S.No.	No. of Hopes	Actual Time(sec.)	Predicted Time(sec.)	Error (%)
1.	1	0.654	0.632	0.0336
2.	1	0.567	0.532	0.0617
3.	2	0.768	0.675	0.1215
4.	2	0.845	0.745	0.1183

4.1. SIMULATION ENVIRONMENT

Entire system is simulated in MatLab R2012a and time series neural network toolbox to generate the outputs. Although, the attributes of the network is already defined in table 2, yet here once again we discuss about the results produced. The entire simulation work has been performed in MatLab software. The wired network has several attributes and these parameters are inputs to Neural Network and figure 5 shows the basic tools to start our simulation.

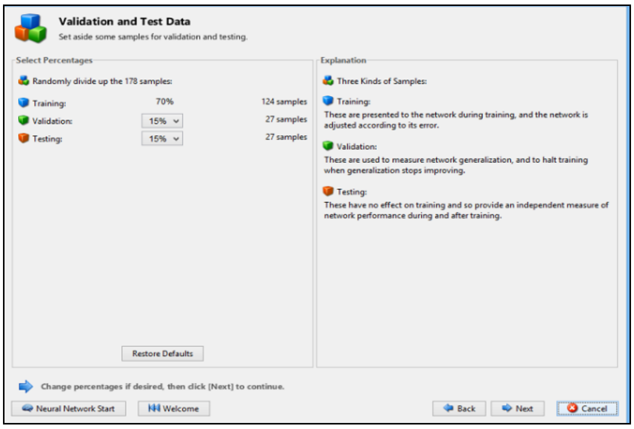

Figure 5. Validation frame for the attributes

4.2. SIMULATION RESULT

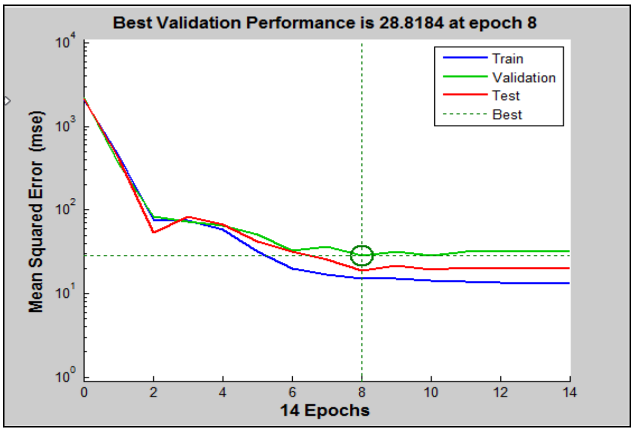

Figure 6.Validation performance after 8 epochs

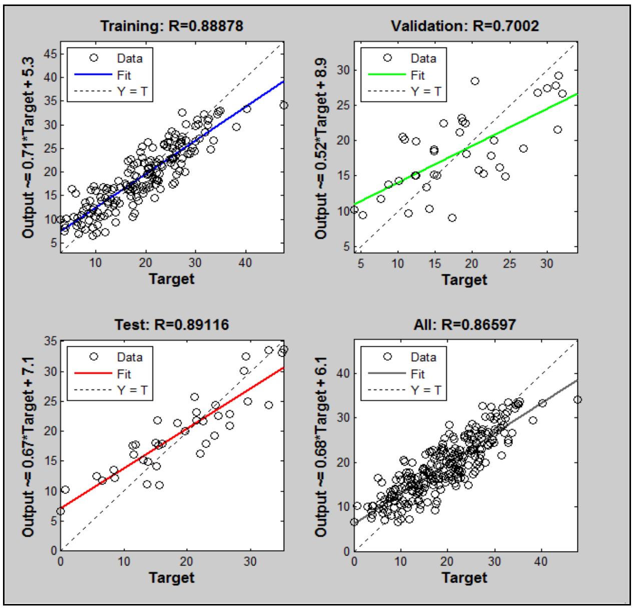

Figure 7. Regression performance

5. Conclusion And Future Work

The wired network is widely used for online connectivity and Internet users require high speed and low delay. Therefore this system is remained to verify in high speed Internet network. Although, this system has been verified and tested in small network of wires, yet it requires the testing in higher bandwidth network. The given work is proposed for the Ethernet and low level wired connection.

6. Acknowledgement

This paper is supported by our mentor and guide Sachin Kumar Saxena, Member of Editor Team of International Journal of Advancements in Research & Technology (ISSN 2278-7763), IJOART Publication, USA. The website reference for it is given here to check http://www.ijoart.org/editorialTeam.shtml and College of Engineering Roorkee, Roorkee for providing the basic resources. The authors would like to thank Dr. Garima Krishna and Dhaneshwar Kumar in order to support in Matlab simulation work and compare this work to real environment.

References

[1] Ruey-Maw Chen, Yueh-Min Huang, "Competitive neural network to solve scheduling problems", *0925-2312/01/$ - see front matter _ 2001 Elsevier Science B.V.*

[2] Tran Xuan Truong, Le Hung Lan, "Congestion Control In TCP/IP Differentiated Services Network Using Neural Network", 9781424499533/11, 2011.

[3] Harvey B. Newman, Iosif C. Legrand, *A Self-Organizing Neural Network for Job Scheduling in Distributed Systems*, Kent State University, Kent, OH, USA, Doctoral Thesis, 2003.

[4] Martin H. Luerssen, *Character Recognition with Neural Networks*, School of Informatics & Engineering Flinders University of South Australia GPO Box 2100 Adelaide 5001 Australia.

[5] Heikki N. Koivo, *NEURAL NETWORKS: Basics using MATLAB Neural Network Toolbox*, February 1, 2008.

[6] V. Venkatesa Kumar and K. Dinesh, *Job Scheduling Using Fuzzy Neural Network Algorithm in Cloud Environment*, Bonfring International Journal of Man Machine Interface, Vol. 2, No. 1, March 2012.

[7] Martin Hagan, *Neural Network Design ECEN 4120 (Call # 45307) / ECEN 5120 (Call # 45308), July 2007*.

[8] Klimis Symeonidis, *Hand Gesture Recognition Using Neural Networks*, School of Electronic and Electrical Engineering On August 23, 2000.

[9] Qinghui Lai1, Haiye Yu1, Haitao Chen, Yong Sun, *Prediction of Total Power of Agricultural Machinery Using Artifical Neural Networks, IEEE International Conference on Computing, Control and Industrial Engineering, 2010*.

[10] Dibyajyoti Guha, S.S. Pathak, *Load Balancing using Past Information of Queue*, Kent State University, IEEE 2010.

