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

Sharma Astha#,
Department of CSE,
GRD-IMT, Uttarakhand, India

Gupta Nikita$
Department of IT,
COER, Uttarakhand, India

Kochhar Radhika@
Department of IT,
COER, Uttarakhand, India

Abstract: - The wired 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 while communicating from one node to another node. 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 TCP-friendly congestion controls. Entire work is predicted by Neural Network because it can process multidimensional inputs concurrently and conventionally we had linear regression but now we have NN to give the output more precisely.

Keywords: Neural Network, Congestion Control, Round Trip time, TCP/IP.

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. After training, the neural network operates as an adaptive controller under changes in TCP dynamics [10]. 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. During network training, the weights and the bias are iteratively updated until they reach their optimal values [5].

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 [9]. 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 [4]. There are mainly two methods for training neural networks: a back-propagation-trough-time algorithm, feed forward learning algorithm.

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. 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. The activation function used of these feed forward neural networks is a sigmoid function as in equation depicted [8].

$$f(x) = \frac{1}{1 + e^{-\lambda x}}$$

Figure 2. Sigmoid activation function

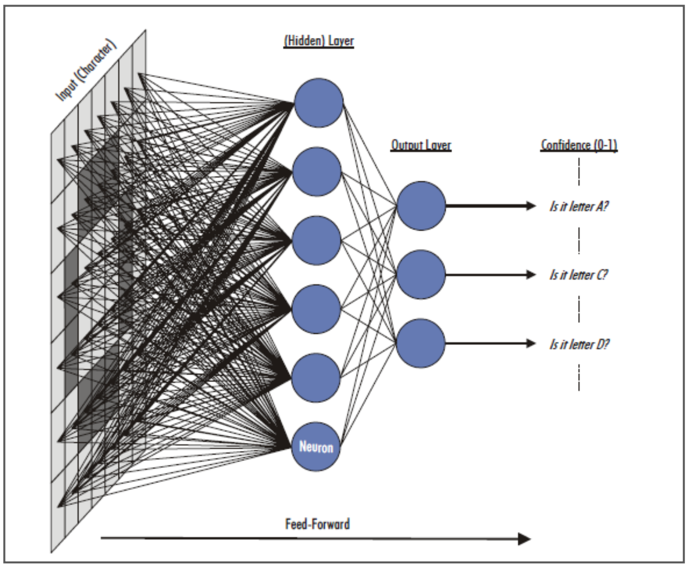

Figure 1. Architecture of Neural Network

2.3. TRAINING

The neural network designed in Section II must be trained to optimize a TCP network performance measure. 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.

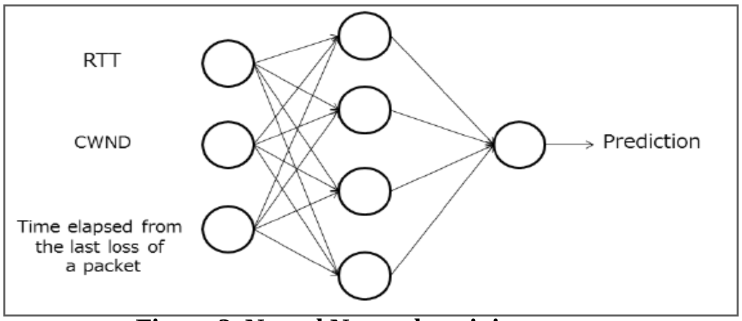

Figure 3. Neural Network training

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.

» net=newff([-1 2;0 5],[3 1],{'tansig' 'purelin'},'traingdm');

Note that the first input [-1 2; 0 5] is the minimum and maximum values of vector p. We might use *minmax(p)* , especially for large data set, then the command becomes:

»net=newff(minmax(p),[3 1],{'tansig' 'purelin'},'traingdm');

We then want to train the network net with the following commands and parameter values (or we can write these set of command as M-file).

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 [7]. 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. 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.

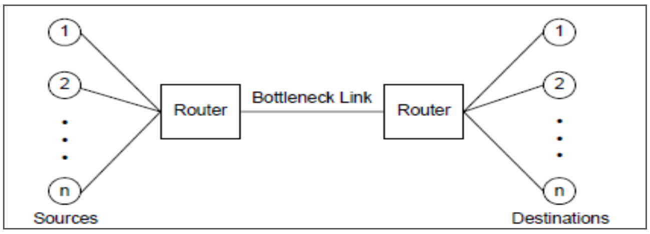

Figure 4. Routes of packets

One of the major problems for most communications networks lies in defining an efficient packet routing policy. 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. During the simulation, the control agent executes at a polling interval, monitoring the traffic and making decisions. Files are created for each node to keep track of that node's data. During each run of the network simulator, the files are extended with the new data from the latest interval [3]. The most important part of the control agent is the neural network prediction module. For our agent implementation, we used a single hidden layer, feed-forward neural network. This was a compiled application, so wrappers were needed to control the input and output dealing with the neural network [6]. The wrapper program was written in C and is called after the data files are updated by the simulator. The simulator halts until the C program finishes.

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. 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 [2].

5. Simulation Environment

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. This screen is the demonstration of the real world scenario. Similarly, figure 6 illustrates the input parameters to the neural network.

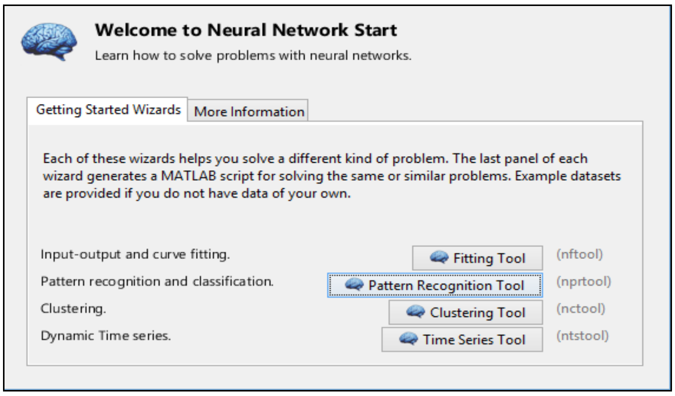

Figure 5. Neural Network MatLab Environment

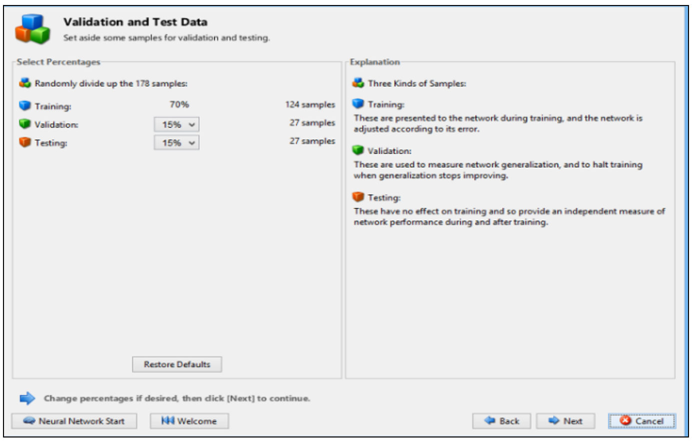

Figure 6. Matlab Validation Toolbox

5.1. SIMULATION RESULT

The results are explained in table 1, which compares the actual values and predicted values of our system. The actual values are extracted from the wired network in geographical area and predicted values are defined by the validation graph of neural network. The validation, training and testing part is accomplished by MatLab only in order to turn around the feasibility of entire simulation work. One can easily understand the deep root of this system once will go through the figure 7, which the first and foremost screen shot of our simulation work.

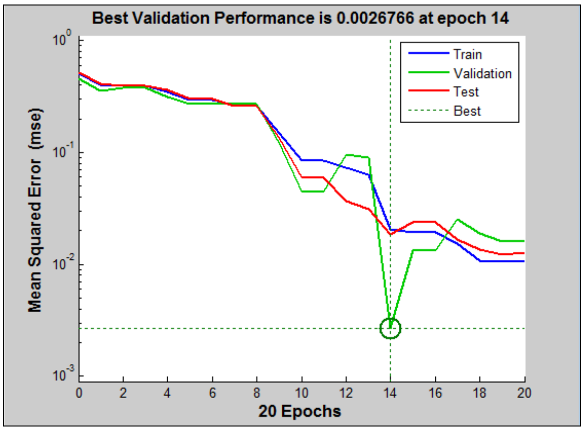

Figure 7. Graph representing training, validation and testing results after 20 epochs

Table 1. Actual and Predicted values

S.No.	Actual value	Predicted Value	% error
1.	0.345	0.322	0.067
2.	0.237	0.227	0.042
3.	0.546	0.487	0.112
4.	0.532	0.456	0.143

Errors are calculated by subtracting Actual values by predicted values and divided by actual value as illustrated in Table 1, column 3.

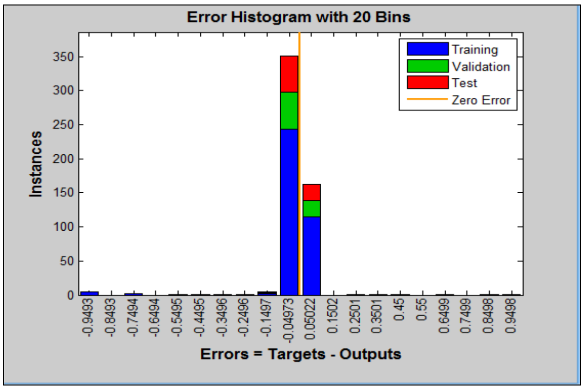

Figure 8. Error report for training, validation and testing

6. Conclusion And Future Work

The wired network is widely used for online connectivity and Internet users require high speed and low delay [1]. 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.

7. 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.

