

An Ann Simulation of Single Server with Infinite Capacity Queuing System



Sivakami Sundari M, Palaniammal S

Abstract: Successful formulation of queuing models depends on arrival rate, nature of waiting in queues, type of service and customer leaving the system depends on type of arrival, nature of service, number of servers deputed, type of queues, number of customers approaching for service in the system and delay. Kendall notations are popularly used for designating the queuing models like M/M/C/E/D. Various mathematical models have been developed to solve the queuing problem analytically. However solving queuing models with power of computers is the new area of research and this work intends to develop single server infinite capacity queuing system using Artificial Neural Network(ANN). The results of simulation are compared with that of analytical method.

Keywords : Artificial Neural Network, Infinite capacity, Queuing System, Single server.

I. INTRODUCTION

Queuing theory involves issues related to waiting for service and queuing. Queues are formed based on the nature of the service and when resources are limited. Queuing theory deals with optimizing the use of resources so that with minimum resources maximum output is possible. In many real time systems the demands of customer arriving, demands on resources and waiting for the service may be termed as queuing system. The schematic of a queuing problem is shown in Fig.1. Queuing theory based mathematical analysis is regularly used in tollgate, telephone industry, software industry, call centers, hospital, airports etc. The purpose of the queuing analysis is how minimum resources effectively used to attain maximum outputs. Queuing simulation is based on the idea of queuing theory and based on which resource allocation is planned and executed during the process of different fields. Fig.2 shows the schematic of a queuing model. Queuing modeling used to find constraints if any in servicing customers and calculates how much time the customer is waiting in the queue. It there by estimates waiting time and there by optimize resources based on the business strategy and plan. In this work an ANN approach is implemented to arrive at solutions for single server infinite capacity queuing system and compared the results with traditional mathematical method.

II. QUEUING THEORY

Queuing theory is used to mathematically analyse a given system waiting for service, and provide critical information about the system working with existing resources and to design new service systems. This theory is useful in calculating the sufficiency of resources and its capacity to serve to satisfy the needs of the customer, by minimizing the waiting time at the same time with minimum cost. The theory of queuing problems worked out with the concept of probability studies using random phenomena. Such phenomena involves and appear in almost all the fields like telecommunication, Physics, engineering, industry, and many other fields. Queuing theory first analysed by AK Erlang in 1913[1] to study telephonic calls and is widely used in second world war operations by Kendall[2] to take decisions on war operation and implemented successfully.

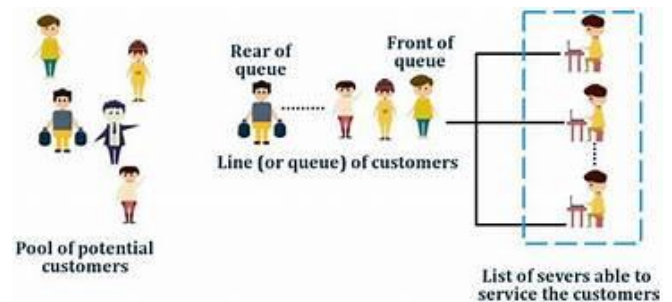


Fig.1 Schematic of Queuing Problem

III. QUEUING MODEL

In this work Queuing model of single server infinite capacity queuing is analytically solved using Kandal[2] and Little's formulas[3]-[9]. Inputs for solving the queuing models considered are (i) Arrival rate λ , (ii) service rate μ , (iii) No of customers in the system exceeds 'N', (iv) size of the population 'n', and the results of queuing models are calculated using the following equation from (1) to (8) .

(i) Traffic intensity of the queue

$$\rho = \frac{\lambda}{\mu} \tag{1}$$

(ii) Probability that the system on service is idle

$$P_0 = \frac{1-\rho}{1-\rho^{N+1}} \tag{2}$$

(iii) Average number of customers waiting in the system

$$L_s = \rho - \frac{(N+1)\rho^{N+1}}{1-\rho^{N+1}} \tag{3}$$

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Sivakami Sundari M*, Department of BSH, GIET University, Gunupur, India. Email: senthilsivakami@gmail.com

Palaniammal S. Principal and Professor, Department of Mathematics, Sri Krishna Adithya College of Arts and Science, Kovaipudur, Coimbatore, India. Email: splvlbresearch@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>



(iv) Effective arrival Rate of customers

$$\lambda' = \mu(1 - P_0) \tag{4}$$

(v) Average number of customers in the queue

$$L_q = L_s - \frac{\lambda'}{\mu} \tag{5}$$

(vi) Average waiting time of the customer in the system

$$W_s = \frac{L_s}{\lambda'} \tag{6}$$

(vii) Average waiting time of the customer in the queue

$$W_q = \frac{L_q}{\lambda'} \tag{7}$$

(viii) Probability of 'n' customers in the system

$$P_n = P_0 \rho^n \tag{8}$$

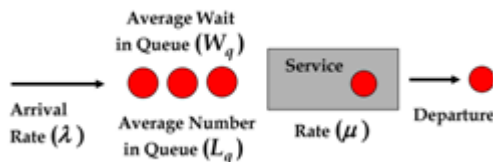


Fig.2 Schematic diagram of Queuing Model

IV. ARTIFICIAL NEURAL NETWORK (ANN)

ANN models gain more attention owing to their application in various fields like mathematics, transportation, weather and market trend forecasting etc. The ANN model consists of three layers mainly input layer, output layer and hidden layer[10]. The schematic diagram of ANN model is shown in Fig.3. Each layer is connected by neurons with interconnection from input to output layers. The number neuron to be fixed each layer is depends on the nature of the problem, for example number neurons in the input layer is fixed by number on inputs required, to solve the problem using analytical method. The number of output layer depends on our requirement of what we would like to calculate. Hidden layer number of neurons are arbitrary fixed and by which number of neuron results good approximation of the given problem is art of the research in ANN modeling. The best suited number of neurons in the hidden layer for the particular problem with given input and output is selected to predict the unknown data.

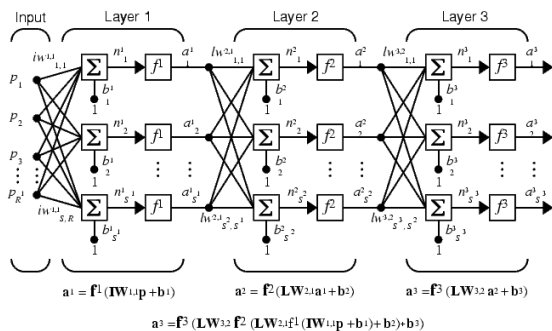


Fig.3 Schematic diagram of ANN Model

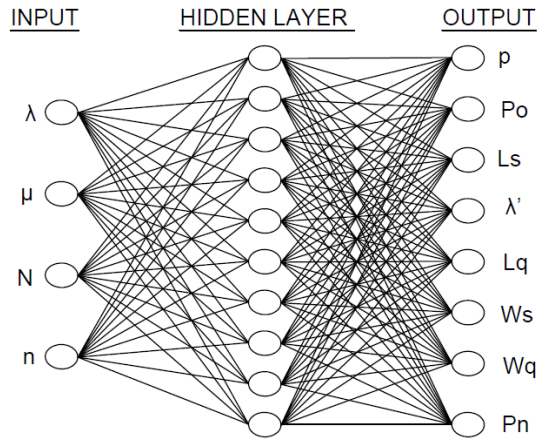


Fig.4 Architecture of Queuing Model

In this work input layer is modeled with four neuron and output layer is fixed with eight neurons. Number of hidden layers started with one hidden layer and subsequently increased to obtain better accuracy of the solution. MATLAB is used to model the ANN and the program to run the training, supervision and testing is given below.

```
function net = fitwithnet(p,t)
%FITWITHNET Creates and trains a neural
network to fit input/target data.
%
% [NET,PS,TS] = FITWITHNET(P,T) takes:
% P - RxQ matrix of Q R-element input samples
% T - SxQ matrix of Q S-element associated
target samples
% arranged as columns, and returns these
results:
% NET - The trained neural network
% PS - Settings for preprocessing network
inputs with MAPMINMAX.
% TS - Settings for postprocessing network
outputs with MAPMINMAX.
%
% For example, to create an network with
this function:
%
% load housing
% net = fitwithnet(p,t);
%
% To test the network on the original or new
data:
%
% y = sim(net,p); % Apply network
% e = t - y; % Compare targets and outputs
%
% To reproduce the results you obtained in
NFTOOL:
%
% net =
fitwithnet(mmlinfifoINPUT,mmlinfifoI
NPUT);
```



```
% Create Network
numHiddenNeurons = 28; % Adjust as desired
net = newff(p,t,numHiddenNeurons);
% Division of Samples
% (Delete these lines to use default
settings of 0.6, 0.2 and 0.2.)
net.divideParam.trainRatio = 0.60; %
Adjust as desired
net.divideParam.valRatio = 0.20; % Adjust
as desired
net.divideParam.testRatio = 0.20; % Adjust
as desired
% Random Seed for Reproducing NFTool
Results
% (Delete these lines to get different
results each time this function is called.)
rand('seed',1307970494.000000)
net = init(net);
% Train Network
[net,tr] = train(net,p,t);
% Simulate Network
[trainOutput,Pf,Af,E,trainPerf] =
sim(net,p(:,tr.trainInd),[],[],t(:,tr.trainInd));
[valOutput,Pf,Af,E,valPerf] =
sim(net,p(:,tr.valInd),[],[],t(:,tr.valInd));
[testOutput,Pf,Af,E,testPerf] =
sim(net,p(:,tr.testInd),[],[],t(:,tr.testInd));
% Display Performance
fprintf('Train vector MSE:
%f\n',trainPerf);
fprintf('Validation vector MSE:
%f\n',valPerf);
fprintf('Test vector MSE: %f\n',testPerf);
% Plot Regression
figure
postreg({trainOutput,valOutput,testOutput}, ...
{t(:,tr.trainInd),t(:,tr.valInd),t(:,tr.testInd)});
```

ANN modeling has done using feed forward Back propagation algorithm.70% of the available data obtained out of mathematical model is used to train the model and remaining 20% data used for testing and 10% for validation.

	1	2	3	4	5	6
1	0.0830	0.1000	4	1.8000	0.1110	0.3333
2	0.1000	0.1250	5.7000	2	0.3333	0.4000
3	10	10	10	10	10	10
4	3	3	3	3	3	3

	7	8	9	10	11	12
1	0.0830	0.0830	0.0666	0.0166	0.3300	6
2	0.1660	0.2500	0.1666	0.0500	0.6600	8
3	10	10	10	10	10	10
4	3	3	3	3	3	3

	13	14	15	16	17	18
1	0.2000	0.0660	0.1660	0.2500	0.0500	0.2500
2	0.5000	0.1000	0.5000	0.3330	0.6666	0.5000
3	10	10	10	10	10	10
4	3	3	3	3	3	3

	19	20	21	22	23	24
1	0.5000	0.3333	0.0760	6	0.0166	0.8330
2	0.6660	0.4166	0.2500	8	0.0330	0.9000
3	10	10	10	10	10	10
4	3	3	3	3	3	3

Table- I: Input layer data for four neuron set for training

	1	2	3	4	5	6
1	0.8300	0.8000	0.7018	0.9000	0.3330	0.8333
2	0.1700	0.2000	0.2982	0.1000	0.6670	0.1668
3	0.1411	0.1600	0.2093	0.0900	0.2221	0.1389
4	0.3111	0.3600	0.5075	0.1900	0.8891	0.3057
5	4.8824	4	2.3529	9	0.4993	4.9970
6	4.0524	3.2000	1.6512	8.1000	0.1663	4.1638
7	58.8235	40	0.5882	5	4.4984	14.9925
8	48.8235	32	0.4128	4.5000	1.4981	12.4925
9	0.4746	0.4096	0.2425	0.6561	0.0123	0.4821
10	0.5718	0.5120	0.3456	0.7290	0.0369	0.5785
11	5.8824	5	3.3529	10	1.4993	5.9970
12	0.5908	0.4852	1.2027e-15	0.0165	0.0039	0.2195
13	0.7118	0.6065	1.7139e-15	0.0183	0.0117	0.2634
14	0.0264	0.0215	0.0086	0.0349	1.1194e-05	0.0269

Table- II: Input layer data for four neuron set for training

TrainLM is used to train the ANN and Feed Forward Back propagation algorithm is used for supervised learning with bias. Combination of hardlim, purelin, logsig and tansig transfer functions used between input layer to hidden layer and output layer to meet the target. Number of neuron in the hidden layer is arbitrarily used starting from 6 neurons to 28 neurons and the results of the testing data compared with that of analytical values.

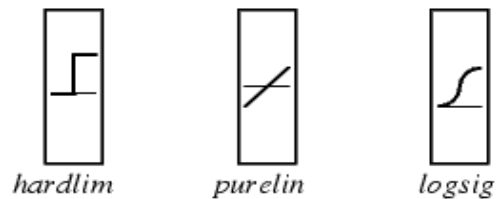


Fig.5 Transfer functions used in ANN Model

Training of ANN model is done based on the two approaches. One approach is to find out the best model based on the set goal value, and the other approach is for the set goal value the number of epochs the network taking to train the model to achieve the target. If the set target is achieved with small number of epochs the best network modeling is finalized to proceed further for testing and validation. The training curve generated for the ANN model is given in the fig.6 where the goal set is zero error and achieved in the training process is 0.0006

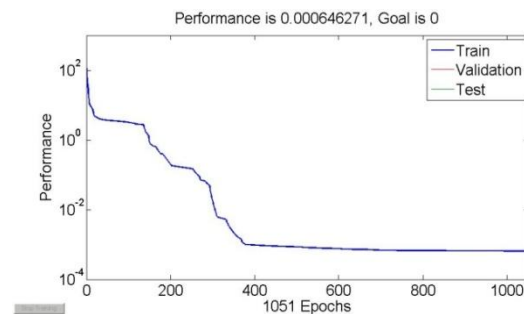


Fig.6 Training curve of seven neurons

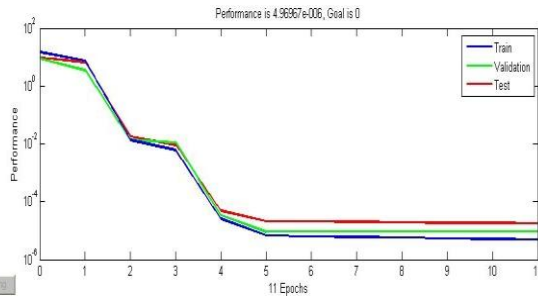


Fig.7 Training, validation and testing curve ANN

V. RESULT AND DISCUSSION

Fig.7 shows the trend of training , validation and testing of the single server infinite capacity queueing model using ANN. The simulation process stabilized and reached the target goal of zero error within 11 epoches which indicates the ANN modelled is well weighted with the input and target data sets.

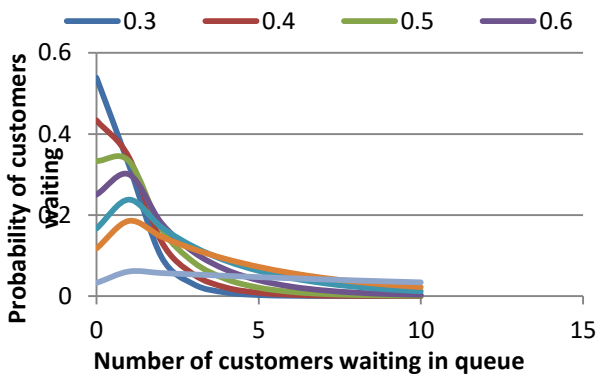


Fig.8 Simulated single server Queuing Model

The probability of number of customers waiting in the queue is simulated and for various intensity values from 0.3 to 0.9 probability of customers waiting is simulated and is shown in the Fig.8. Mathematically calculated values tabulated in Table.III

Table- III: Results of Mathematical Modeling of a single server queue

Traffic Density	No of customers waiting for service in the queue										
	0	1	2	3	4	5	6	7	8	9	10
0.3	0.5385	0.3231	0.0969	0.0291	0.0087	0.0026	0.0008	0.0002	0.0001	0	0
0.4	0.434	0.3426	0.1352	0.0534	0.0211	0.0083	0.0033	0.0013	0.0005	0.0002	0.0001
0.5	0.3333	0.3333	0.1667	0.0833	0.0417	0.0208	0.0104	0.0052	0.0026	0.0013	0.0007
0.6	0.25	0.3	0.18	0.108	0.0648	0.0389	0.0233	0.014	0.0084	0.005	0.003
0.7	0.1667	0.2381	0.1701	0.1215	0.0868	0.062	0.0443	0.0316	0.0226	0.0161	0.0115
0.8	0.1176	0.1858	0.1467	0.1158	0.0914	0.0722	0.057	0.045	0.0355	0.028	0.0221
0.9	0.0323	0.0605	0.0567	0.0532	0.0498	0.0467	0.0438	0.0411	0.0385	0.0361	0.0338

VI. CONCLUSION

An Artificial neural network model successfully developed to solve single server infinite capacity queueing problem. The developed model is trained, validated and tested with the analytical model results. From the error analysis we found that the ANN model is very well fitting

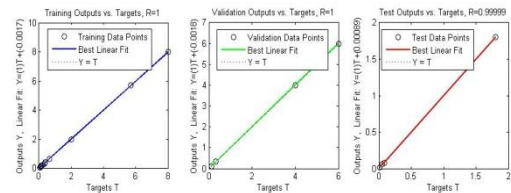


Fig.8 Error analysis of single server ANN Queuing Model

Thus, it is checked that the training procedure was successful. The network converged and produces the correct target outputs for the four input vectors.

The ultimate result of the ANN network is checked for its accuracy by regression analysis against the predicted output value to the actual inputs given in the input layer values[7]-[10]. MRE, correlation coefficient and RMS values are calculated in evaluating the ANN. Fig.8 shows the error analysis and the R value is almost equal to 0.999 means the simulated value and target value is almost matching and the error is minimum. Equation number “8” defines the calculation of R value

$$R(a, p) = \frac{\text{cov}(a, p)}{\sqrt{\text{cov}(a, a)\text{cov}(p, p)}} \tag{8}$$

Cov(a, p) - covariance between a and p sets that, refer to the actual output and predicted output sets

Cov(a, a) & Cov(p, p) - auto covariance of a and p sets, correspondingly.

The correlation coefficient lies between -1 and +1. R values closer to +1 specify a stronger positive linear relationship while, R values closer to -1 indicate a stronger negative relationship. The mean relative error, which shows the mean ratio between the errors and experimental values, is evaluated by

$$MRE (\%) = \frac{1}{N} \sum_{i=1}^N \frac{100(ai-pi)}{ai} \tag{9}$$

with that of analytical model and capable of predicting the target parameters for the given input data with minimum negligible error.

REFERENCES

1. EBrockmeyer, HLHalstrom, AJensen, "The life and works of AK Erlang", Trans. Danish Acad. Tech. Sci.(2) (1948).
2. D.G.Kendall, "Some Problems in theory of queues", Journal of the Royal Statistical Society , vol. 13, 1951, pp. 151-157 and 184-185.
3. J.Abate, W.Whitt, "Transient behavior of the M/M/1 queue: Starting at the origin", Queueing Systems , vol 2, 1987 pp. 41-65.
4. D.J.Daley, "The Correlation Structure of the Output Process of Some Single Server Queueing Systems", The Annals of Mathematical Statistics, vol. 3,1968, pp.1007-1019.
5. L.Flatto, "The waiting time distribution for the random order service M/M/1queue", The Annals of Applied Probability, vol.7, 1997, pp.382-409.
6. G.Koole, A.Mandelbaum," Queueing Models of Call Centers: An Introduction", Annals of Operations Research, vol. 113(1-4), 2002, 41-59.
7. R.Kannapiran Palvannan, Kiok Liang Teow, "Queueing for Health care", Journal of Medical Systems, vol. 36(2), 2012, 541-547.
8. Oblakova, A., Al Hanbali, A., Boucherie, R.J., "An exact root-free method for the expected queue length for a class of discrete-time queueing systems", Queueing Syst, vol. 92, 2019, pp. 257-292
9. Talak, R., Manjunath, D. & Proutiere, A., "Strategic arrivals to queues offering priority service", Queueing Syst., 92, 2019, pp. 103-130.
10. M.T.Hagan, H.B.Demuth, M.Beale, "Neural Network Design", PWS Publishing Company , Boston,MA,1996.

AUTHORS PROFILE



Sivakami Sundari M, Department Of Bsh, Giet University, Gunupur, India Has 13 Years Of Teaching And Research Experience Has Published Articles In International Journals And Conferences.



Dr. S. Palaniammal Principal and Professor Department of Mathematics, Sri Krishna Adithya College of Arts and Science, Coimbatore, India has 32 years of rich teaching and research experience. She has published 167 papers in International and National journals/conferences and also visited countries like Dubai, Malaysia, Singapore, Saudi Arabia and Thailand. She authored 8 books to her credit, Prentice Hall of India, New Delhi published two of her books titled Random Processes & Probability and Queueing Theory and the book titled Linear Algebra Calculus and its Applications was published by McGraw Hill Education (India) Pvt. Ltd, Chennai.