User Support System for Selecting the Fastest Serving Queue to avoid Queuing at Tolls Using BPNN

S. Ashok Kumar

Associate Professor & Head of Computer Science, AVP College of Arts and Science, Tirupur

Abstract- **Toll gates are always a place known for its congestion and queuing. This is a well known highly competitive problem in all parts of the world for the highways department. Various solutions such as automatic toll system, electronic card systems and open systems are in existence which has reduced or eradicated the congestion problem. This paper is a plan of proposing a support system for the vehicle drivers. The proposed intelligence system will help the users (drivers) who pass by the tolls, in selecting the fastest serving queue of a particular toll gate. The previous data of processing the number of vehicles at a particular lane are taken into consideration. The previous data are used to train and produce an intelligence system which will help the users to choose the fastest serving queue. This paper presents only a rough set of data which is taken as an example and a methodology is shown how such a system may be implemented using neural network intelligence system.**

Index Terms- **Backpropogation, Congestion, Neural Network Queuing, Tolls.**

I. INTRODUCTION

Tollway is commonly called as toll. It is the road which is either a private or public road. Here fee or tax or toll is collected which the vehicle passes by the passage termed as lanes. Tolls are collected in particular place which is termed as toll booths or toll houses or toll plazas, or toll gates. The tax or toll amount will vary depending on the type of the vehicle or weight of the vehicle or based on the number of axles.

The toll roads include various time consuming constraints such as queuing and congestion while paying the toll costs which is considered as the biggest problem. Further in the tolls during peak period of traffic the queuing and congestion problem still gets bigger[1]. Many automated systems for toll paying has been implemented at various levels have minimized the problems. This paper is a study on implementing the intelligence concept in choosing the fastest servicing queue of the lane.

II. RELATED WORKS

A detailed literature survey was made in the area of tollgate and neural network and its algorithms. Oskar Kärrman and Linnéa Otterlind (2017) explored artificial neural network to find the performance of the travel time and traffic volume using Taguchi method[11]. Yiqian Huang, Liang Chen, Yanwen Xia, Xiuliang Qiu (2018) has designed a expressway toll station by analyzing daily traffic flow using fuzzy BP-Neural model[10]. Finkenzeller. K (2012) had proposed a technique in order to reduce the waiting time at the toll gates. This method looks to be very secured and authentic. This method also stores the passers data and vehicle numbers for further operations. Hiroyuki Yoshikawa, (2007) had proposed a vehicle management system, which will get rid of the user dissatisfaction in toll collection system at the time of traffic congestion. Bean Micheal (1994) had proposed the system of toll collection for the need of better road way systems in order to avoid the congestion. Don. F. Hadwiger (1990) had proposed a system that is used to collect the toll tax ordered by the processor. This method has the demerit of having higher time consumption rate.

III. TOLL MANAGEMENT SYSTEM

Toll tax is collected at various places for different reasons like 1. To meet the cost of construction and rate of interest to the expense occurred 2. To meet the day to day operation expenses that occurs on tolls and 3.To meet the road repairs and maintenance of the roads, sign boards etc. Highways and roads are also not an exception. In earlier stages, the construction, extension, maintenance and operating costs of highways, roads, bridges and tunnels were collected directly or indirectly[7].

Toll system can be broadly categorized into two types. They are 1. Open toll system and 2. Closed toll system. This paper deals for the closed toll system where the passenger has to wait in the queue for his processing. The processing is based on various methods of toll collection such as a. Manual toll collection, b. Automatic toll collection like coins and tokens and c. Electronic Toll Collection (ETC) points like Fast tag systems. Here the processing time is an important factor and it is calculated as the difference between the time of the entry of a vehicle to the toll queue in any lane and the time it leaves the toll area.

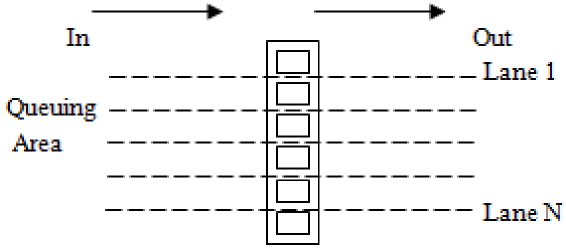

Fig 3.1 Toll Plaza Geometry

3.1 Toll Processing

Tlls are single line input stream where the vehicles can serve only one vehicle at a time. The implementation of multiple serving input streams (multiple lanes), will initiate parallel servicing and queuing avoidance. The incoming vehicles have to be staked in any one of the queuing line for its service to cross towards the out area. Here the FIFO (First-in First-out) methodology is implemented. The following is the java pseudo code initialization that shows the FIFO technique implementation in stacking and processing the input.

```
public Stack(int s)
{
    maxSize = s;
    stackArray = new long[maxSize];
    top = -1;
}
public void push(long j)
{
    stackArray[++top] = j;
}
public long pop()
{
    return stackArray[top--];
}
public boolean isEmpty()
{
    return (top == -1);
}
public boolean isFull()
{
    return (top == maxSize - 1);
}
```

The push in stack is equivalent to the vehicle entering and selecting a lane. When the service is finished at the service place they will be poped out of the lane. Here the service is done in FIFO order and the element is moved out of the service at the earliest possibilities. The following pseudo java code describes the final level how the vehicle is pushed in and poped out of the stack in FIFO order.

```
static void showpush(Stack stack1, int a)
{
    stack1.push(new Integer(a));
}
static void showpop(Stack stack1)
{
    Integer a = (Integer) stack1.pop();
}
```

IV. BPNN ALGORITHMS

BPNN (Back Propagation Neural Network) algorithms are used in solving many intelligent systems[3]. Various versions of the backpropogation algorithms are available. This chapter is planned to give an outline of the various steps involved in the back propagation algorithm. The Neural Network (NN) proposed contains three layers. Namely input layer, hidden layer, and output layer. During the network training phase, the sample training data is feed into the input layer. From the input layer the data is transmitted to the hidden layer and then from there the result is passed to the output layer. This process is called as the forward pass of the back propagation NN algorithm.

In the forward pass of the algorithm, each node in the hidden layer will get the input from all the nodes from the input layer. Concurrently appropriate weights are multiplied with it and then it is summed up. The output of the hidden node will be the non-linear transformation of the resulting sum. Likewise each node in output layer gets input from all the nodes from the hidden layer, which are then multiplied with the appropriate weights and then summed up. The output of the node will be the resulting sum. Now the output values of the output layer will be compared with the original target output values and if the target output values are same then the error between the actual output values and the target output values will be calculated and then it will be propagated back to the hidden layer. This is called the backward pass of the back propagation algorithm.[4]. Following code is the pseudo code of backpropogation NN algorithm[5][6]

The following code is the initial weights assigned to the neurons present in the input layer where the weights will be calculated and moved towards the hidden layer.

```
init_weights_from_inputs_to_hidden_layer_neurons(self, hidden_layer_weights)
{
weight_num = 0;
for h in range(len(self.hidden_layer.neurons))
for i in range(self.num_inputs)
if not hidden_layer_weights
self.hidden_layer.neurons[h].weights.append(random.random())

else
self.hidden_layer.neurons[h].weights.append(hidden_layer_weights[weight_num])
weight_num += 1
}
```

The below code is the initial weights assigned to neurons in the hidden layer which will be propagated to the output layer.

```
init_weights_from_hidden_layer_neurons_to_output_layer_neurons(self, output_layer_weights)
{
weight_num = 0;
for o in range(len(self.output_layer.neurons))
for h in range(len(self.hidden_layer.neurons))
if not output_layer_weights
self.output_layer.neurons[o].weights.append(random.random())
else
self.output_layer.neurons[o].weights.append(output_layer_weights[weight_num])
weight_num += 1
}
```

The following code shows how the training set calculates the error function and return it back for the next calculation for the training. This is a very important area in the training of data.

```
calculate_total_error(self, training_sets)
{
total_error = 0;
for t in range(len(training_sets))
training_inputs, training_outputs = training_sets[t]
self.feed_forward(training_inputs)
for o in range(len(training_outputs))
total_error += self.output_layer.neurons[o].calculate_error(training_outputs[o])
return total_error
}
```

V. PROPOSED SYSTEM

Neural network uses a different idea in solving a given problem instead of the idea used by the conventional methods. The conventional methods will use the algorithmic steps to solve the problems. These methods will use a set of instructions in order to solve the problem [2]. To solve any problem the steps of the algorithms should be known to us in advance. This characteristic limits the problem solving capability of the regular conventional methods in solving a given problem. Neural networks process the given information alike the human brain processes the information. The neural network consists of a very large number of highly interconnected processing elements called neurons which works in parallel to solve a specific problem [2]. A neural network intelligence system is proposed to the users where the users can select the best lane which has the fast processing speed. A sample data for six months are taken assuming a toll gate with five lanes. The sample data of processing the vehicles at each lane on the toll is taken as the input. The graph in Fig5.1 shows the sample data taken and their processing at five lanes.

The sample data is used to train the network. The network consisted of input layer, a hidden layer and an output layer with necessary neurons. During the training process the goal was set to 0.01 accuracy. The network was able to nearly predict the right lane which has processed the highest number of vehicle processed.

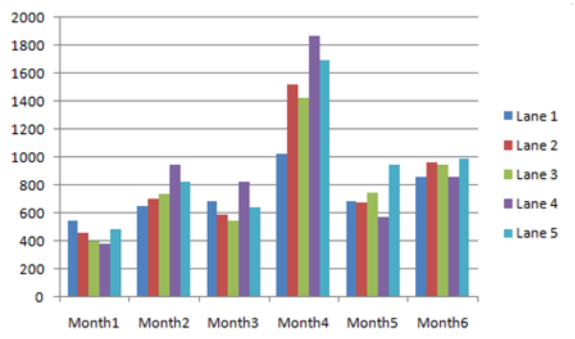

Fig 5.1 Sample data plot for five Lanes

With the considered example it is found that out of all other lanes, lane4 is the fastest serving queue. The special time delays, stoppage of processing of vehicles for a certain period due to various reasons, delay due to human errors, closure of lanes for various reasons for a specified time interval were are not considered during the training process. Adding of those data sets and characteristics to the input layer will make the network to predict the better solution or nearest possible solutions. This intelligence system will help the toll passers to presume the fastest processing lane and which will help them to queue themselves in the fastest processing track.

VI. CONCLUSION

This paper is a study on implementing the intelligence in the toll processing area. This will help the vehicle passers to choose the fastest servicing queue of the lane which may in turn reduce the traffic congestion. The training is based on the previous processing data of the tolls on queue services at a particular toll plaza. Various constraints are not considered in training the network. Including those factors will give better results which will be taken up for the future work. This new facility will provide a solution for reduced waiting time on the tolls. In future, the intelligence system can also be planned and implemented to the open toll systems.

REFERENCES

[1] Hai Yan, William H.K Lam, Optimal road tolls under conditions of queuing and congestion, Transportation Research Part A: Policy and Practice, Volume 30, Issue 5, 1996, Pages 319-332, ISSN 0965-8564

[2] S. Ashok kumar, G.M. Nasira and T.S.S. Balaji, Neural Network Implementation for Integer Linear Programming Problem, 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 18Improved Precision Recommendation Scheme by BPNN Algorithm in O2O Commerce, 2013 IEEE 10th International Conference on e-Business Engineering

[3] Jamshid Nazari and Okan K. Ersoy , Implementation of back-propagation neural networks with MatLab, Purdue University School of Electrical Engineering 1992

[4] http://wwwold.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node22.html

[5] https://mattmazur.com/2015/03/17/a-step-by-step- backpropagation-example

[6] A Feedback-Based Dynamic Tolling Algorithm for High-Occupancy Toll Lane Operations Guohui Zhang, Yinhai Wang, Heng Wei, Ping Yi, The national academics of sciences, engineering, medicine, https://trrjournalonline.trb.org/doi/abs/10.3141/2065-08

[7] DeCorla-Souza, Patrick,, Fair Lanes : A New Approach To Managing Traffic Congestion, The national academics of sciences, engineering, medicine, Vol.8 No.2,pp 5-13

[8] S. Nandhini, P. Premkumar, Automatic Toll Gate System Using Advanced RFID and GSM Technology, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Issue 11, November 2014

[9] Hiroyuki Yoshikawa, Vehicle management system, Patent application publication May 3, 2007, US2007/01 00687A1

[10] Yiqian Huang, Liang Chen, Yanwen Xia, Xiuliang Qiu, Design of Expressway Toll Station Based on Neural Network and Traffic Flow, American Journal of Operations Research, Vol.08 No.03(2018), Article ID:84913,17 pages

[11] Oskar Kärrman and Linnéa Otterlind, Highway tollgates traffic prediction using a stacked autoencoder neural network, https://www.chalmers.se/highway-tollgates-traffic-prediction-using-a-stacked-autoencoder-neural-network.aspx