Optimizing Traffic Signals At an Intersection Using ML

Min Gyu Kim
Department of Statistics, Seoul National University, Seoul, Korea
Email: *aksen0@snu.ac.kr,

***Abstract*—In this paper multilayer perceptron(MLP) is proposed to optimize traffic signal system at an intersection. With a simulation model for vehicles passing through an intersection, we sought to find the signals that minimize the sum of square of the time spent by each vehicle. With this simulation model, a data set of the distribution of the vehicle labeled with optimal signal were created. With this labeled data we have made a MLP that derives the optimal signal when the distribution of the vehicle is given in advance.**

I. Introduction

Every day a lot of time is wasted on the road just for waiting the traffic signal. If we could cut this time even just a little, it would be a big gain.

All cars will someday become autonomous(Although it is ideal to assume an autonomous vehicle, it can be applied to the reality by collecting data from sensors that measure traffic flow). Then they can be connected to a central processing system for optimum efficiency. All cars move simultaneously like an organism, achieving optimal efficiency in transportation. Since the city's road network is made up of multiple intersections and most of the traffic jam occurs at the intersection, we confined the problem at one intersection. The goal of this research is to find an optimal traffic signal system at an intersection to minimize the total time spent by vehicles.

II. Related Works

There have been many studies to collect traffic data and optimize traffic signals. ITS(Intelligent Traffic System) is a system that measures and manages traffic flow. ITS uses methods such as bluetooth, GPS and video recognition to measure the traffic flow. Also there are many algorithms to solve this problem. The most common ones are genetic algorithms [1] and reinforcement learning [2].

III. Problem Definition

It is not yet possible to obtain autonomous vehicles, so data can not be obtained directly from experiments. Even if we replace autonomous vehicles as an ideal driver, it will be difficult to obtain sufficient data. Therefore, we made a simulation model that measures the time for each vehicle to cross the intersection, when the distribution of the vehicles and the traffic signal are given in advance.

Vehicles entering the intersection can be divided into 4 straight and 4 left turns as shown in Fig 1 (In this model, we ignore the right turn and crosswalks). Also, in order for the vehicle to move without collision, only eight flows are allowed as the traffic signal, as shown in Fig 2. The traffic signal changes every 10 seconds to have a total of 4 signals for 40 seconds. (If this model is created successfully, reducing the time interval will make the traffic signal system change flexible, close to continuous change) It is assumed that the distribution of the vehicle follows the Poisson distribution (which is a common assumption for these kinds of problems). In order to make data available for the input of MLP, it is necessary to make discrete data of a certain length, so the arrival time is rounded. The input data is the number of vehicles which arrive at time t, where $t = 0, 1, ...39$. The example of movement of cars for each signal is expressed in table 1. Also, after 40 seconds have passed, the remaining cars that haven't got out of the intersection pass through at last. We measure the time spent for each vehicle and look for signals

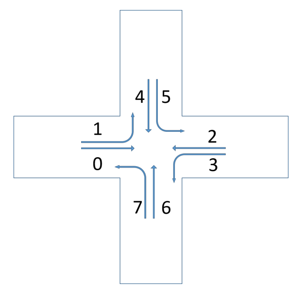

Fig. 1. 8 traffics

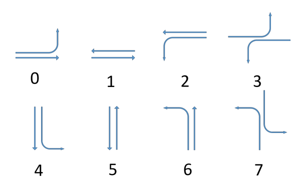

Fig. 2. 8 signals allowed

Time(s)	Signal	movement of cars
0 to 10	7	5,7(from fig 1) are moving, others are waiting
10 to 20	5	4,6(from fig 1) are moving, others are waiting
20 to 30	2	2,3(from fig 1) are moving, others are waiting
30 to 40	0	0,1(from fig 1) are moving, others are waiting
end	-	all the remaining cars pass through

TABLE I
TABLE 1 : EXAMPLE OF SIGNALS AND CORRESPONDING CAR MOVEMENTS.

that minimize the sum of the square of the time value. The reason for squaring is to penalize the algorithm that causes a particular car to wait too long.
The problem expressed as equations is as follows.
$t \in N^{320}$(N is nonnegative integers) : distribution of vehicles
$s \in [0,1]^{32}$: signals
$S(t,s)$: the sum of squared time for vehicles passing thorough intersection with distribution t and signals s
$G(t) : N^{320} \to [0,1]^{32}$ G is a three layer perceptron that gets the distribution of vehicles and generates optimum signal.
Target :

$$g(t) = argmin_s S(t,s) \tag{1}$$

Objective : to find G that minimizes

$$\sum_{t \in test\ data} ||G(t) - g(t)||_2^2 \tag{2}$$

IV. Solving Approach

The possible number of signal in one time interval(10s) are 8, and the signal changes every 10 seconds from the entire 40 seconds, so there are a total of 8^4 possible numbers. We apply the simulation model to all of these signals to find the optimal signal for a specific distribution of vehicles. Repeating this, for each randomly generated distribution of vehicles, the optimum signal is labeled. When composing MLP, the input is the number of vehicles entering at 0,1, ..., 39 seconds(the length of input is 40 * 8 = 320. 40 for time, 8 for each flow in figure1). This is represented by $t_{i,j}$, where i is each flow in figure 1 and j is the each time(second). In this case, each signal is considered to be a vector having a length of 8 and each element having 0 or 1. For example, the signal number 2 is [0,1,0,0,0,0,0,0]. Now, we can create a MLP with a total of 4 * 8 = 32 outputs, representing the corresponding four signals per hour as vectors of length 8 each. This is represented by $o_{i,j}$, where i is each signal in figure 2 and j is the each time(j = 0,1,2,3). To sum it up, we used three layer perceptron with 320 inputs and 32 outputs. GDM was used and Rectified Linear Unit was used as an activation function.

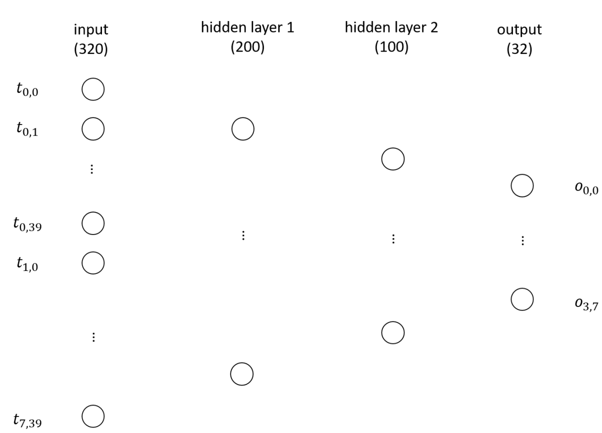

Fig. 3. Three Layer Perceptron used in this paper

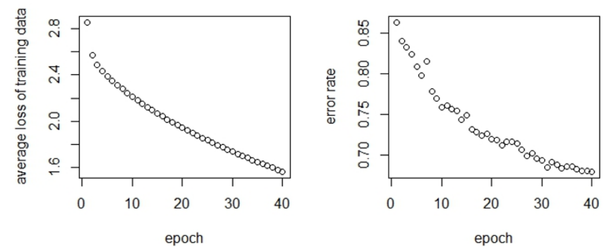

Fig. 4. Plot 1. average loss per epoch / Plot 2. error rate per epoch

V. Evaluation

47818 training dataset was collected, and 2926 dataset was used to test. The error rate of test data is 67.98%. Since there is $\frac{1}{8^4}$ chance of getting hit with arbitrary signal, this error rate shows that MLP can learn to optimize the signal from the data. In fact, we can see that the average loss from training data and error rate from test data are decreasing at figure 4.

The error rate can be decreased if we get more training dataset and fine-tune the hyperparameter of this MLP.

VI. Conclusion

In this study, we used MLP to optimize a simple traffic model at an intersection.

The direction for further research can be divided into two. The first one is to make the simulation model closer to reality. We have created a simulation model that assumes many things arbitrarily to make model simple. But in reality it will be more complicated than that. Crosswalks should be added, the size of each car is different, and the safety distance must be maintained. If the car is crowded, there may be a car that has not run out until the signal is over. We might also think of a variant that minimizes the reaching time of certain important cars (such as ambulances) by giving a weight to each car. In addition, we set time limit in this study but in practice, it is necessary to calculate the optimum signal for the constantly coming cars. Most of all, We should get data from real intersections to validate simulation model [3].

Improving neural net is a second one. Although MLP was used in this study, there are lots of variation of neural nets that is superior than MLP. It is known that reinforcement learning or recurrent neural networks such as LSTM are more efficient in analyzing these kinds of data [4]. Unlike MLP, RNNs can use their internal state to process sequences of inputs. It would be able to increase the accuracy by using these kinds of neural nets.

References

[1] K. T. K. Teo, W. Y. Kow, and Y. K. Chin, "Optimization of traffic flow within an urban traffic light intersection with genetic algorithm," in *2010 Second International Conference on Computational Intelligence, Modelling and Simulation*, Sep. 2010, pp. 172–177.
[2] E. Walraven, "Traffic flow optimization using reinforcement learning." Masters thesis, Delft University of Technology, 2014.
[3] M. J. Vandeman, *Is Traffic Signal Synchronization Justifiable?*, 1994.
[4] J. Hochreiter, Sepp; Schmidhuber, *"Long Short-Term Memory". Neural Computation.* Prentice-Hall Englewood Cliffs, NJ, 1997, vol. 9, no. 8.