Capstone Proposal - Traffic Lights Planning

TSUNGEN SU
April 15st, 2017

Proposal

Domain Background

In the “train the smartcab to drive” project, the smartcab learned how to reach its destination safely and efficiently by using Q-learning. Traffic lights, on the other hand, is another important factor for the traffic system. Traffic light control system is pre-programmed by trained traffic engineer and is used to control traffic flow. However, if the traffic light control system can be self-learned for many situations that might be able to reduce traffic congestion and improve traffic efficiency.

The similar research paper “Reinforcement Learning for Adaptive Traffic Signal Control” suggested utilizing reinforcement learning, specifically on Q-Learning, to dynamically optimize a standard four-way intersection stop light. However, it only focus on one four-way intersection.

This project aim at minimising car waiting time by a self-learned traffic light planner in a finite space. For example, the10x10 grid space in a four-way intersection the traffic light control system should stop the direction which has less car in waiting and most cars are allowed to passing. The smartcab project simulator is reused in this project for traffic situation. The all intersection are concerned in the area, not just in one four-way intersection.

Problem Statement

The problem with traditional traffic light control is it is running on regular interval basis regardless traffic situation. It has only two actions which are green light on north-south and red light on east-west or red light on north-south and green light on east-west. The task is simple, switch them in a time interval. However, to improve traffic flow that requires human intervention. It is to be said, it is not intelligent enough.

What can be done in order to make it intelligent? By saying intelligent, is to set a goal to find a way that improves the problem. To make it measureable, first thing is to build a model of given problem:

Giving a 10x10 grid square area, the line to be thought of roads, the intersection to be thought of 4 way intersection. The input and output are defined as following:

Input Xi: the number of cars on the road in the given the position. For example, x1=5. There are 5 cars on the street 1.

Output Yi, the action of traffic light is taken in an intersection. 0: Green-NS and Red-EW, 1: Red-NS and Green-EW

After modelled it, the question is how can machine learning do to improve the situation?

Datasets and Inputs

This project doesn't require any dataset to be processed. The smartcab project simulator is reused and modified for this project, all cars and its destinations for training and testing are all randomly generated to simulate the real world situation.

To collect data in our environment the snapshot of the environment is needed. The time of capture the snapshot of the given state would be moment of light switching. Think it as a decision point, it is important that we know in what circumstances that agent had made a decision to take an action. The input data is the array of number of cars on ordered road. For example, x1 would be the number of cars in road 1.

Since state is consist of cars and roads, so that the state space is the same as input spaces.

Solution Statement

The traffic light control system is pre-programmed and it sometimes doesn't seem to be reasonable that cars have to stop for a certain amount of time even there is no car crossing in the other direction. For example, a car stop at intersection because of red light is on and it has to wait for 90 seconds to pass. The 85 seconds can be saved if the traffic lights turns green in 5 seconds.

The traffic light control is also blind and require human intervention in a situation when traffic is serious congested, and it doesn't have bird eyes to see a better way that can make situation better. For example, the traffic is serious congested, but heading north cars are more than the cars that heading east. Allowing green light a little longer for north direction cars can save times than equally treated.

In general, the traffic lights is controlled to be changed in the fixed amount of time regardless the situation on the road. In order to make traffic more fluent, the two different light switching time is designed to allow traffic light planning agent gaining control of it. For example, use short traffic light time if there is no car on waiting.

The traffic light control agent is designed to learn the combination of traffic lights' actions in the finite space area with random generated cars driving on the road. The states of the road situation can be infinite, since cars can be as many as space allowed.

The Q learning with Neural Network is therefore to be applied for the traffic light agent. The inputs are the numbers of cars in driving road of the area, and the outputs are the actions to be taken on each intersection. Use Neural Network can approximate the best policies for states. Traffic light agent is trained with random generated cars driving on the road for certain amount of time. For example, the 10 x 10 grid space is created to have 100 intersections and 11*11 roads for car to drive on. Cars can have many as space allowed and driving in random direction. It is obvious use lookup table of Q learning is not an efficient way to approximate the situation.

The Q learning with Neural Network can be found at http://outlace.com/Reinforcement-Learning-Part-3/.

Since the traffic is only thing agent cares about, so the cars can start at any position in the grid and begin randomly walking. If the actions we choose is 2 sec and 4 sec switching time, then the decision points could be at the moment of 2 sec. The following are simple steps that capture the flow of code:

- Random generate a number of cars and put onto the random position of road, the car objects are cached.
- At two seconds interval, we first check the state of car at ordered roads. This would be the input values to our network.
- Generate actions for the intersections. This is where we train and test our network.
- Based on our actions the cars can do the action to react traffic lights. Update cars' state.
- Calculate rewards by checking car's state. Reward function is simply accumulate the cars' state, giving running car a +1 and stopped car a -1.
- Backpropagation.

Benchmark Model

The maximum time of traffic light can be used as the benchmark model to compare with traffic light agent performance. For example, if we use 2 sec and 4 sec action time in our model then the 4 sec is used in the benchmark model. The benchmark model performance can be measured by average the running fixed amount of traffic light switch time in 10 trials.

Evaluation Metrics

Since the traffic flow is only thing that traffic light agent care about. The running cars and stopped cars can be used as performance measurement. The more cars running on the road without stop for the traffic light is the goal for this project.

We use the average number of running cars on the road on specific of time to measure the performance. For example, giving 10 cars on the road, in the 5x5 grid space, the score is 52 (5.2/10x100) if there are at least 5.2 cars are on the run per minute. The score is between 0 and 100.

Project Design

The simulator of "Train smartcab to drive" project will be used and modified for the sake of demonstration.

Before put on test, the traffic light agent will need to be trained. Training will be in the random number of auto generated cars driving on the fixed square area for a certain amount of time. For example, we train the agent in 5 trials, each trial takes 30 minutes to run. The number of cars is the random generated number and is changed every 10 minutes. The running cars are consider a positive rewards and stopped cars are negative. The total rewards is the sum of running cars and stopped cars number.

The neural network is built by taking grid position as input data and traffic light actions as output data. The combination of neural network is the objective that this project going to investigate.

After neural network is trained then we can run the test trials to compare with benchmark model.