Advanced Engineering Forum *Online: 2011-12-22*
ISSN: 2234-991X, Vols. 2-3, pp 91-95
doi:10.4028/www.scientific.net/AEF.2-3.91

Development of a Neural Network Based Q Learning Algorithm for Traffic Signal Control

Li Bi Fu, Kil To Chong
School of Electronics Engineering, Chonbuk National University, Korea
E-mail : kitchong@jbnu.ac.kr

Keywords: Q learning, neural network

Abstract. As one kind of reinforcement learning method, Q learning algorithm has already been proved to achieve many significant results in traffic signal control area. However, when the state of Markov Decision Process is very big or continuous, the computation load and the memory load will become very big and can not be solved then. Therefore, this paper proposed a neural network based Q learning algorithm to solve this problem known as "Curse of Dimensionality". This new method realized generalization of conventional Q learnig algorithm in huge and continuous state space as neural network is a very effective value function approximator. Experiment has been implemented upon an isolated intersection and simulation results show that the proposed method can improve the traffic efficiency significantly than the conventional Q learning algorithm.

Introduction

Nowadays more and more attention has been focused to the application of reinforcement learning on real time traffic flow control. This method considers learning to be a trial-and-error process. Sutton proposed a developed learning algorithm for non-deterministic Markov decision processes [1]. Lu Shoufeng applied table Q-learning to dynamically control the traffic signals at an isolated intersection [2]. Wei Wu also developed a coordinated urban traffic signal control approach based on multi-agent reinforcement learning [3]. Marco Wiering developed a Multi-Agent reinforcement learning method for traffic light control of six adjacent intersections [4].

The advantage of reinforcement learning is that no mathematic model of the controlled object needed, the system can perceive the varying condition and self-adaptively adjust the control policy in order to respond to traffic conditions. It makes the control object optimal due to its self-learning ability. However, the problem known as "Curse of Dimensionality" which was supposed by Bellman in his book at 1961 is still exist when we apply the traditional reinforcement learning algorithm. When the state of Markov Decision Process is very big or continuous, the computation and memory load will become very big and can not be solved then. On the other side, in the traditional Q learning algorithm, Q value is updated in the form of table record, the effeciency of this kind of learning is relatively slow, which will directly influence the performance of the controller [2].

To solve the problem of "Curse of Dimensionality", and realize the high effeciency of reinforcement learning in huge and continuous state space, Value Function approximation based learning method has been widely researched and applied in recent years. This paper proposes a neural network based Q learning algorithm to find a dynamic planning of traffic flow control and we proposed a heuristic knowledge based Q learning algorithm.

This paper was organized as follows. Section 1 introduced Q learning algorithm and the improvement. The experiment and simulation results were studied in Section 2. Finally, section 3 provide and conclusion of our research.

1 Q Learning algorithm and its improvement

1.1 Traditional Q learning algorithm

Q learning algorithm was first supposed by C.Watkins in his thesis at 1989, which was applied to the iterated computation of value function in the Markov Decision Process. The iteration equation is:

$$\begin{aligned} Q(s_t,a_t) &= Q(s_t,a_t)+\alpha[r(s_t,a_t)+\gamma \max Q(s_{t+1},a_{t+1})-Q(s_t,a_t)] \\ &= (1-\alpha)Q(s_t,a_t)+\alpha[r(s_t,a_t)+\gamma \max Q(s_{t+1},a_{t+1})] \end{aligned} \tag{1}$$

(s_t,a_t) is state-action pair at time t of the Markov Decision Process, s_{t+1} is the state at time $t+1$, $r(s_t,a_t)$ is the immediate reward at time t, $\alpha > 0$ is the learning factor.

1.2 BP Neural Network based Q learning algorithm

In this paper, we use the BP learning algorithm to realize the learning process of output layer and hidden layer weights. The structure of the neural network can be seen in Fig.1.

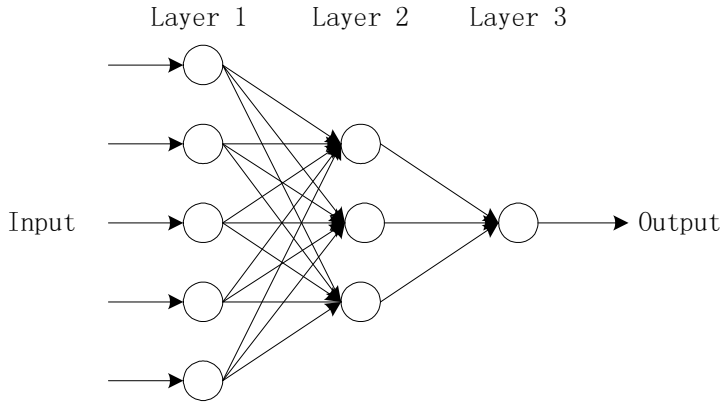

Fig. 1. Structure of three layer neural network

Supposed that there are n neurons in hidden layer and the output of hidden layer in ith neural network is y^i, value of hidden layer weight of output layer is w^i, and then we can get the evaluate result of the action value function as:

$$Q(s,a_i)=(w^i)^T y^i=\sum_{j=1}^{n} w_j^i y_j^i \tag{2}$$

The transfer functions of hidden layer and output layer are defined as follows:

$$f_1(net_1)=\frac{1}{1+e^{-(net_1)}} \tag{3}$$

$$f_2(net_2)=net_2 \tag{4}$$

Thanks to the research in [5], we know that learning process here is not like ordinary supervised learning where we can learn from (input, output) example. Here we're not presenting constant $(x,Q^*(x))$ examples to the network but instead we are learning from estimates of $\max Q(s_{t+1},a_{t+1})$, according to the iteration equation in (1), we know from output of the neural network to the learning object is as followed:

$$Q(s_t,a_t)\rightarrow r(s_t,a_t)+\gamma \max Q(s_{t+1},a_{t+1}) \tag{5}$$

where $\max Q(s_{t+1}, a_{t+1})$ is an estimate, which may come from a different network, but it is the maximal value of the resulting state. The weights in the network are updated to minimize the following quadratic performance error measure:

$$E=\frac{1}{2}e^2(t)=\frac{1}{2}[r(s_t,a_t)+\gamma \max Q(s_{t+1},a_{t+1})-Q(s_t,a_t)]^2 \tag{6}$$

The updated algorithms of output layer are derived as follows:

$$\Delta w_t^2=-a_t\frac{\partial E}{\partial w_t^2}=-a_t\frac{\partial \frac{1}{2}[r(s_t,a_t)+\gamma \max Q(s_{t+1},a_{t+1})-Q(s_t,a_t)]^2}{\partial w_t^2} \tag{7}$$

The updated algorithms of hidden layer are derived as follows:

$$\Delta w_t^1=-a_t\frac{\partial E}{\partial w_t^1}=-a_t\frac{\partial E}{\partial (net_1)}\cdot\frac{\partial (net_1)}{\partial w_t^1} \tag{8}$$

Here $a_t>0$ is the learning factor of the neural network at time *t*.

1.3 Heuristic knowledge based Q learning algorithm

In order to improve the efficiency of the learning system, we supposed a heuristic knowledge based Q learning algorithm. Assuming that the saturate flow rate of the traffic is *s*, the arriving flow rate is *q*, and the green time that will be decided is Δt. Then we can calculate the arriving traffic during Δt is $q\Delta t$. And the time for this arriving traffic to disperse is:

$$g=q\Delta t/(s-q) \tag{9}$$

To ensure that the traffic queue can disperse during the green time, we get $g \le \Delta t$. We can calculate the delay time for the green lanes as:

$$d_g=\begin{cases} q_0^g\times g/2 & (\mathrm{g}\le\Delta\mathrm{t}) \\ q_0^g\times \Delta t/2 & (\mathrm{g}>\Delta\mathrm{t}) \end{cases} \tag{10}$$

Here q_0^r is initial traffic in green lanes. So the total delay time during Δt is:

$$D=d_g+d_r=\begin{cases} \dfrac{q_0^g\times g+q_0^r\times\Delta t}{2} & (g\le\Delta t) \\ \dfrac{q_0^g\times\Delta t+q_0^r\times\Delta t}{2} & (g>\Delta t) \end{cases} \tag{11}$$

Finally the average delay for the entire intersection is:

$$H(q_0^g,q_0^r,\Delta t)=\overline{d}=\frac{D}{q_0^g+q\times\Delta t+q_0^r+q\times\Delta t}=\begin{cases} \dfrac{q_0^g\times g+q_0^r\times\Delta t}{2(q_0^g+q\times\Delta t+q_0^r+q\times\Delta t)} & (g\le\Delta t) \\ \dfrac{q_0^g\times\Delta t+q_0^r\times\Delta t}{2(q_0^g+q\times\Delta t+q_0^r+q\times\Delta t)} & (g>\Delta t) \end{cases} \tag{12}$$

Equation (12) is the heuristic function that will be added into the learning system. We can see that this function can evaluate the performance of the decision, given the environment information that we know.

After combining the above two methods together, (1) and (13) can be combined as finally as:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) - \varepsilon H(s_t, a_t)] \tag{13}$$

The improved algorithm can be summarized as follow:

(1) Initialize Q(*s,a*), H(*s,a*);
(2) Observe the state at time t;
(3) Pretend to execute each action, observe each new state and receive each reward *r*;
(4) Updating Q function according to (13);
(5) Choose one action according to:

$$a_t = \arg\max Q_{t+1}(s_t, a_t) \tag{14}$$

(6) Go to step 2.

The performance of this algorithm will be tested in next section.

2 Experiment and result

2.1. Isolated intersection model

This chapter will explain how to apply supposed algorithm into the isolated intersection model. As shown in Fig.2, our method is applied to a traffic intersection that consists of two intersecting roads, each with several lanes and a set of synchronized traffic lights that manage the flow of vehicles.

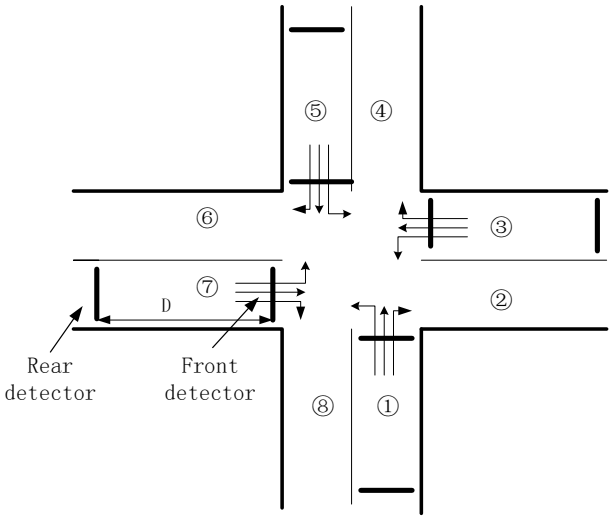

Fig. 2. Sketch map of intersection

Initial traffic data is randomly generated between 0 and 10, and the data for neural network input comply with Poisson distribution:

$$p_k(\Delta t) = \frac{(q\Delta t)^k}{k!} e^{-q\Delta t}, \; k = 0, 1, 2, \cdots \tag{15}$$

The initial weights of neural network are randomly generated between 0 and 0.5. When the entering flow rate is 720veh/h, along with the traffic state updating, the input data is generated resulting from the remaining traffic after the end of previous action and the random arriving traffic which comply with Poisson distribution.

2.2 Simulation result

In order to show the performance of the algorithm in traffic control, testing simulation is performed in this section. Assuming that the arriving rate of traffic is known in 5 minutes, and the optimal traffic signal will be distributed according to the improved neural network and heuristic knowledge based Q learning algorithm (NNHK_Q algorithm). We assume that the arriving flow rate of traffic is 720 veh/h, the simulation was running for 100 cycles and we calculated the average delay at the end of each cycle, the simulation results are shown in Table I:

Table 1 Average delay performance for the whole intersection (720veh/h)

Control Method	Average Delay(s/veh)
Actuated control	29.19
Neuro-fuzzy control	20.72
NNHK_Q control	17.00

3 Conclusion

Traditional table Q learning algorithm was introduced and its disadvantage named "Curse of Dimensionality" was also discussed. In order to solve this problem, BP neural network based Q learning algorithm was studied since neural network has achieved many results in supervision learning field as a popular function approximation. Then in order to improve the performance in traffic control, we proposed a novel NNHK_Q algorithm which was added a heuristic knowledge based function, because heuristic function supports the model information of the environment. Finally, a simulation for such intersection system was carried out, and a comparative study with Actuated control method and Neuro-fuzzy control method was accomplished. The simulation results show that the proposed intersection control mechanism is encouraging and can bring vehicle drivers some benefit by decreasing the average delay.

References

[1] R. Sutton. Learning to predict by the methods of temporal difference. Machine Learning, 1988, 3: pp. 9-44.

[2] Lu Shoufeng, Liu Ximin and Dai Shiqiang. Q learning for adaptive traffic signal control based on delay minimization strategy. IEEE International Conference on Networking Sensing and Control, 2008, pp. 687-691.

[3] Wei Wu, Gong Shufeng and Liu Hongxiu. A coordinated urban traffic signal control approach based on multi-agent. INES'09 Proceedings of the IEEE 13th international conference on Intelligent Engineering Systems , 2009.

[4] Marco Wiring. Multi-agent reinforcement learning for traffic light control. ICML '00 Proceedings of the Seventeenth International Conference on Machine Learning, 2000.

[5] Mark Humphrys. Action selection methods using reinforcement learning. A dissertation submitted for the degree of Doctor of Philosophy in the University of Cambridge, 1997.