УДК 517.977.58

DQN-BASED TRAFFIC SIGNAL CONTROL SYSTEMS

Kondratov Ivan Vladimirovich,
Master student,
Saint Petersburg State University

Abstract. Real-time adaptive traffic control is an important problem in modern world. Historically, various optimization methods have been used to build adaptive traffic signal control systems. Recently, reinforcement learning has been advanced, and various papers showed efficiency of Deep-Q-Learning (DQN) in solving traffic control problems and providing real-time adaptive control for traffic, decreasing traffic pressure and lowering average travel time for drivers. In this paper we consider the problem of traffic signal control, present the basics of reinforcement learning and review the latest results in this area.

Keywords: reinforcement learning, traffic signal control, optimal control.

Urbanization and population growth significantly increase the traffic load in cities. Traffic congestions in large metropolitan areas have long been an everyday occurrence that negatively affects the social environment, reducing labor mobility of the population, especially low-income groups, and generally slowing economic growth, even in crisis and post-crisis periods (for example, before and after the lifting of strict quarantine restrictions during the COVID-19 pandemic) [1]. Reducing the duration or the number of traffic congestions on the roads is an important task, for which, in addition to road planning and infrastructure modifications, traffic signal control (TSC) is used - system that regulate the patterns of passage at intersections and the duration of various traffic signals.

There are several types of traffic signal control systems: deterministic, which uses Webster formula based on historical data and applies pre-determined timing according to current temporal context [2]; semi-dynamic TSC applies control based on current traffic conditions, preferring short-term decision rather than long-term planning [3]. Recently, considerable effort has been devoted to the research and development of fully dynamic traffic signal control systems, that adapt to traffic situation as well as taking into consideration historical data and predictions based on it.

Various approaches were used to develop fully dynamic TSC, such as genetic algorithms [4], dynamic programming [5] and reinforcement learning. Dynamic programming is a mathematical optimization method, which is essentially an approach to solving an optimization problem by dividing it into a set of sub-problems and solving them separately, obtaining a final solution based on the results of solving the sub-problems. Genetic algorithms are a family of algorithms based on the principles of natural selection - several sets of parameters for solving the optimization problem (chromosomes) are formed, then the population is formed, the quality of the solution is checked with the members of the population, after that the most suitable members of the population are selected, their parameters are mixed and random mutation (random change of some parameters in chromosomes) is introduced. This process is repeated until a population member is obtained which gives a solution corresponding to the specified criteria. Both dynamic programming and genetic algorithms outperform static and semi-dynamic based TSC [4] [5], but in recent years focus of the research shifted toward reinforcement learning, after it was combined with deep learning.

Reinforcement learning (RL) is the third paradigm of machine learning, which differs from supervised and unsupervised learning. When in the first case, most often a labeled sample is provided for training, where a set of features is matched with its corresponding class label, and in the second case, there is no labeling at all, reinforcement learning is somewhere in-between. RL is based on Markov decision process, and operates with set of abstractions: agent, environment and reward. Agent observes current state of the environment, takes action and gets reward based on change of state of the environment, with the goal of developing policy that will maximize cumulative reward (sum of all rewards through every step from start to terminal state of the environment). In this sense, it can be interpreted as labeled data set, but reward which servs as labels are non-stationary (depends on current timestamp and state) and changes with the environment, that makes it significantly different from classic supervised learning paradigm [6].

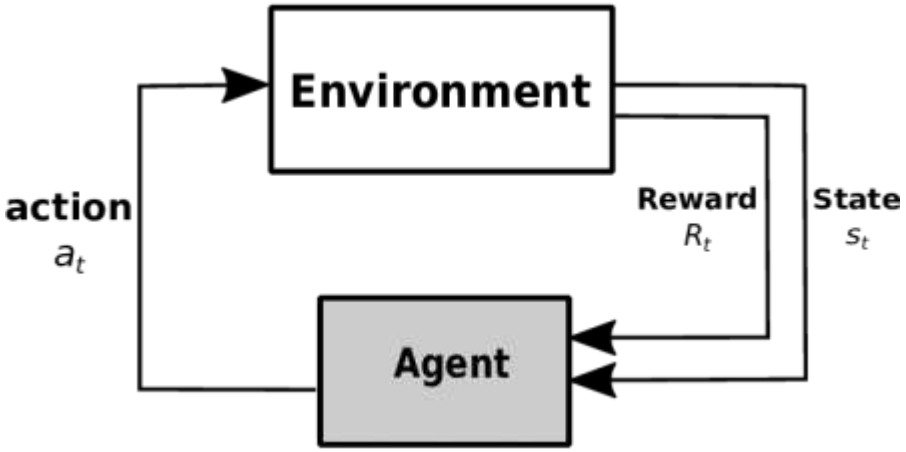

Figure 1. Reinforcement learning schema [6]

One of the most popular algorithms in RL is a Q-learning algorithm [7]. In essence, by solving Bellman optimal policy equation it is possible to obtain optimal policy - conditional distribution of actions over states. When agent samples actions from this distribution in process of learning, he will come to maximal cumulative reward in terminal state. Authors of Q-learning algorithm proposed method of learning through maximizing Q-values, which represent value of particular action observing particular state to a maximization of cumulative reward. Algorithm works as follows: initializing (oftentimes with random values) Q-table, where each possible action corresponds to a row, each possible state corresponds to a column, and table cells contains Q-values for corresponding state-action pairs; after table was initialized agent begin to take actions according to epsilon-greedy policy, which is designed to ensure exploration of the environment (agent takes either random action with probability of epsilon or action that will maximize Q-value for this step with 1-epsilon probability) and updates Q-value based on difference of them for current and previous steps. Ideally, the learning process ends when the algorithm converges, i.e., when the difference in cumulative rewards received at different training phases becomes extremely small, but an infinite learning process is theoretically possible. Q-table, obtained as a result of training and is the optimal strategy, it represents conditional distribution – choosing action that will results in maximal Q-value in turn will lead to maximal cumulative reward [8].

Deep-Q-Learning (DQN) is a modification on Q-learning presented in [9], that combines convolutional neural networks (CNN) and Q-learning architecture. In many cases, it is not always possible to store Q-values in tabular format, especially in case of large number of possible states and actions. Authors proposed to use CNN as an approximator for Q-values and by doing this overcome main Q-learning dimensionality problem. Using CNN in RL paradigm presents significant challenges: as stated previously, temporal dependence of labeling makes CNN instable and varying training data set makes convergence very difficult. Authors develops several modifications to Q-learning algorithm to overcome these challenges. Firstly, DQN uses two CNN to estimate Q-values: target network, which updates its weights every 200 learning iterations and local network, that updates weight in every iteration. DQN also uses replay buffer: to train CNN, algorithm samples from random subset of episodes from learning history. These two modifications allowed DQN to become useful in various fields, from playing games [9], grid systems control [10] or TSC.

Various papers show that using DQN for constructing fully-dynamic TCM is beneficial. In [11] authors define the intersection as environment, represented by grid-like structure. Each grid cell corresponds car-sized segment of intersection, to guarantee that no two vehicles can be held in the same grid. For each car present at the current time is assigned value based on current speed and position in grid, for other cells value set as zero. Based on various grids matrix is constructed, that represents dynamic of cars changing positions, arriving and departing. Agent supervising intersection (crossing of two 6-lanes highways) decides duration of changes of traffic lights. Yellow light is special and shall be included between green and red to ensure drivers safety. Agent choses which light to activate in the next four timestamps, 5 seconds each. As a result of proposed model, evaluated on real-time traffic simulator SUMO [12], authors achieved 20% reduction in average waiting time for the drivers over fixed-time rules and conventional adaptive traffic signal control. Novelty of the approach is the design of action space, that can allow control of sequence of lights, rather than individual selection. This approach lowers number of actions needed to be taken, positively impacting convergence of the model.

Authors of [13] propose a way of controlling traffic lights placed in number of intersections simultaneously using multi-agent variation of DQN. Environment for each agent is presented by its own intersection, described with number of cars and pressure parameter – difference between number of arriving and departing cars on the lanes of the intersection. Actions are defined as set of traffic lights schemas for intersection of two six-lanes highways. Reward defined as a pressure level on intersection after traffic light schema has been activated. DQN-based agents share neural network parameters on episodic basis between each other. Evaluated on real world data obtained from Manhattan, model showed improvement

over static and semi-dynamic TSC and ML-based solutions in average travel time for a driver and throughput of intersection by 10-15%.

In paper [12] presented a way of controlling intersections of various types, some containing ramp connections to suburban highway, some connected to centroids. In total model controls 8 intersections that differs in number of lanes, number of moving phases. Every intersection presented as a matrix, which represents grid with car-sized cells. Cells that correspond to a position of a car are filled with 1, other cells filled with 0. Each agent observes one intersection, choosing actions from pre-defined set of phases of movement that describes traffic light behavior in given state. Reward designed as difference in the cumulative waiting time of the vehicles in the queue for current and previous step. Agent penalized when the vehicle is queued, and when waiting vehicle is discharged agent get a reward. At every control step state matrices are collected. For coordination purposes, agent collects also data on the upstream and downstream intersections are considered (3 intersections in total). In total agent observes and controls its own intersection, also obtaining information about nearby traffic. Simulation scenario based on traffic in suburban area of Seminole County, Florida. Evaluation showed, that in peak hours (from 7 to 9 AM) average travel time decreased by at least 10% and average delay decreased by 50% comparing to currently used TSC in that area. It worth noting, that average number of stops per vehicle increased on average by 22%.

Reinforcement learning is a novel way of solving control problems, and its implementation in TSC showed, that it can improve situation on the road, decrease travel time which in turn will help to mitigate social issues caused by traffic congestion, increase social mobility and stimulate economic growth of the given area.

References

1. Moyano, A., Stępniak, M., Moya-Gómez, B. et al. Traffic congestion and economic context: changes of spatiotemporal patterns of traffic travel times during crisis and post-crisis periods // Transportation. – 2021. – pp. 1-24.

2. B. Yin, A. El Moudni, M. Dridi. Traffic network micro-simulation model and control algorithm based on approximate dynamic programming // IET Intell. Transp. Syst. – 2016. – vol. 10. – no. 3. – pp. 186–196.

3. Cools, Seung-Bae, Carlos Gershenson, Bart D'Hooghe. Self-organizing traffic lights: A realistic simulation // Advances in Applied Self-Organizing Systems. –2013. – pp. 45–55.

4. A. Tamimi, M. Abu Naser, A. Tawalbeh, K. Saleh. Intelligent Traffic Light Based On Genetic Algorithm // IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). – 2019. – pp. 851-854.

5. Chen Cai, Chi Kwong Wong, Benjamin G. Heydecker. Adaptive traffic signal control using approximate dynamic programming // Transportation Research Part C: Emerging Technologies. – 2009. – Volume 17. – Issue 5. – pp. 456-474.

6. R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan and D. Matolak. A Machine Learning Approach for Power Allocation in HetNets Considering QoS // IEEE International Conference on Communications (ICC). –2018. – pp. 1-7.

7. Watkins, C.J.C.H., Dayan, P. Q-learning // Machine Learning. – 1992. – №8. – pp. 279–292.

8. Richard S., Andrew G. Reinforcement Learning: An Introduction. Second // The MIT Press – 2018.

9. Mnih, V., Kavukcuoglu, K., Silver, D. et al. Human-level control through deep reinforcement learning // Nature. – 2015. – №518. – pp. 529–533.

10. T. Sogabe et al. Smart Grid Optimization by Deep Reinforcement Learning over Discrete and Continuous Action Space // IEEE 7th World Conference on Photovoltaic Energy Conversion. – 2018. – pp. 3794-3796.

11. X. Liang, X. Du, G. Wang and Z. Han. A Deep Reinforcement Learning Network for Traffic Light Cycle Control // IEEE Transactions on Vehicular Technology. – 2019. – vol. 68. – no. 2. – pp. 1243-1253.

12. Gong, Yaobang, et al. Decentralized network level adaptive signal control by multi-agent deep reinforcement learning // Transportation Research Interdisciplinary Perspectives. – 2019. – vol. 1.