Video Demo: Deep Reinforcement Learning for Coordination in Traffic Light Control

Elise van der Pol [a] Frans A. Oliehoek [a,b]

[a] *University of Amsterdam*
[b] *University of Liverpool*

Abstract

This video demonstration contrasts two approaches to coordination in traffic light control using reinforcement learning: earlier work, based on a deconstruction of the state space into a linear combination of vehicle states, and our own approach based on the Deep Q-learning algorithm.

1 Overview

The cost of traffic congestion in the EU is estimated to be 1% of the EU's GDP [2], and good solutions for traffic light control may reduce traffic congestion, saving time and money and reducing pollution. To find optimal traffic light policies, reinforcement learning uses reward signals from the environment to learn to make optimal decisions. This demo[1] shows the difference in policies for a multi-agent traffic network, between Deep Q-learning (DQN) with transfer planning [6] and earlier work by Kuyer et al. [3].

2 Approach & Experimental Setup

Earlier reinforcement learning approaches to traffic light control relied on simplifying assumptions over the state and manual feature extraction, so that potentially vital information about the state is lost. Techniques from the field of deep learning can be used in deep reinforcement learning to enable the use of more information over the state and to potentially find better traffic light policies.

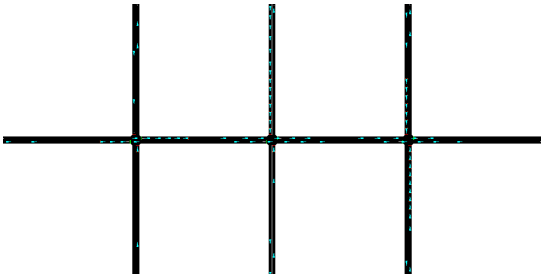
The algorithms compared are the approach used by Kuyer et al. [3], with the adjustment that we learn the joint local Q-value functions, and then use transfer planning with max-plus to coordinate. Similarly, in the DQN approach, we use the DQN algorithm to learn joint local Q-functions, and then transfer planning and max-plus to coordinate. The difference is in the state representation and reward function. In the Kuyer approach, the state is decomposed into a linear combination of lane positions. The reward function penalizes each halted vehicle on the agent's roads. In the DQN approach, a binary matrix of vehicle positions is used, following earlier work [7], and the reward function balances different objectives.

3 Results

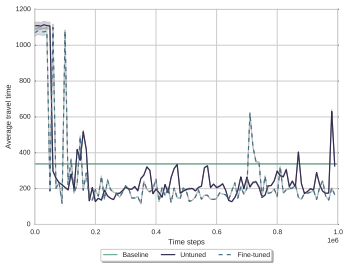
The state is represented as a matrix of vehicle positions, and fed into a convolutional network. We learn the joint Q-value functions between neighbouring agents with the DQN algorithm [4] and use transfer planning [5] with the max-plus coordination algorithm [1] to find an optimal joint action for each traffic network state. The algorithms are compared on amongst others the three-agent scenario in Figure 1a.

[1] http://www.fransoliehoek.net/trafficvideo

To illustrate the behavior of the DQN approach, Figure 1b shows the average travel time for an untuned and a fine-tuned agent, by evaluating the greedy policy at different times during the training process. The best version of the Kuyer algorithm is plotted as a horizontal line.



(a) Three agent traffic scenario: each junction is regulated by a traffic light agent.



(b) Average travel time in the three-agent scenario, for an untuned and fine-tuned DQN agent. The baseline is the Kuyer algorithm.

Figure 1

The video demo shows the behavior of both the DQN approach and the Kuyer approach. The Kuyer agents switch traffic light configurations very rapidly, whereas DQN allows a queue to build up. Because of Kuyer's rapid switching, vehicles are almost always accelerating or decelerating. This is a result of the reward function used, which assigns a penalty for each halted vehicle. In contrast, DQN uses a reward function that balances multiple objectives, one of which minimizes rapid decelerations (emergency stops).

After a few minutes of the simulation the Kuyer agents start getting congested, whereas the DQN agents see less congestion and no rapid switching. This suggests that our approach will be able to handle a higher traffic density. However, near the end of the simulation, the Kuyer agents eventually empty all their roads, but the DQN agents seem 'stuck'. This may be caused by a loss of information when converting from the continuous traffic situation to a discrete position matrix. If the matrix misses some vehicles, or the DQN agent's convolutional filters do not trigger for these vehicles, the result is that the DQN agent behaves as though there are no vehicles on the road.

4 Conclusion

While the DQN approach is promising and finds better policies than earlier work, there are state instances where it fails to take the right action. Thus, more work is needed to increase its robustness.

Acknowledgements

Research supported (in part) by NWO Innovational Research Incentives Scheme Veni #639.021.336. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. ISBN: 0387310738.

[2] Commission of the European communities. *White paper European transport policy for 2010: time to decide*. 2001.

[3] L. Kuyer et al. "Multiagent reinforcement learning for urban traffic control using coordination graphs". In: *ECML-PKDD*. 2008.

[4] V. Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533.

[5] F. A. Oliehoek, S. Whiteson, and M. T. Spaan. "Approximate solutions for factored Dec-POMDPs with many agents". In: *AAMAS*. 2013.

[6] E. van der Pol. "Deep reinforcement learning for coordination in traffic light control". Master's Thesis. University of Amsterdam, 2016.

[7] T. Rijken. "DeepLight: Deep reinforcement learning for signalised traffic control". Master's Thesis. University College London, 2015.