Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Evaluation of Proximal Policy Optimisation in Continual Reinforcement Learning

Kim Nia Nolle

Supervisor: Prof. Vinny Cahill

Co-Supervisor: Wenlong Wang

August 31, 2024

A dissertation submitted in partial fulfilment
of the requirements for the degree of
M.Sc. Computer Science (Data Science)

Declaration

I hereby declare that this dissertation is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent / do not consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. **Please consult with your supervisor on this last item before agreeing, and delete if you do not consent**

Signed: _____ Date: _____

Abstract

Summary of the dissertation

Keywords: this, that, more

Acknowledgements

Thank you to ...

Contents

List of Figures

List of Tables

Acronyms

CL Continual Learning. 1, 3

DQN Deep Q-Learning. 1, 6

i.i.d. independent and identically distributed. 1

MDP Markov Decision Process. 3, 4

POMDP Partially Observable Markov Decision Process. 3, 4

RL Reinforcement Learning. 1, 3–6

SGD Stochastic Gradient Descent. 1

1. Introduction

Despite rising costs and the climate-crisis, driving continues to be one of the most popular modes of transportation. One major problem with the high traffic demands is traffic congestion. The TomTom 2023 Traffic Index has shown that a majority of global cities have seen a rise in travel times in 2023 compared to the previous year, especially during peak hours. Not only does traffic congestion lead to an increase in travel times, but it is also associated with increased greenhouse gas emissions and air pollution (Department of Transport 2023), which in turn are linked to health risks such as cardiovascular and respiratory diseases (World Health Organisation 2022).

In order to curb these issues, effective traffic management is needed. This can be achieved for example by using traffic lights to control the flow of traffic. However, inefficient traffic light cycles can worsen congestion, for example through cross-blocking (when the downstream lane is full so that a vehicle cannot cross the intersection) or green-idling (when a green signal is active even though there are no vehicles crossing in that direction) (Rasheed et al. 2020).

One promising approach to dynamically optimising traffic light cycles is Reinforcement Learning (RL) because no prior assumptions about the systems are required (Liu et al. 2023). Most research in utilising RL for traffic light control is based on Deep Q-Learning (DQN) (Rasheed et al. 2020), which uses Stochastic Gradient Descent (SGD) for optimisation (Mnih et al. 2015). SGD and SGD-based algorithms assume stationary independent and identically distributed (i.i.d.) data distributions (Khetarpal et al. 2022). However, the environment in traffic control can be considered non-stationary because, for example, the distribution of traffic demand varies over time (higher demand during rush hours etc.). Blindly applying these approaches in non-stationary settings can lead to biased optimisations and catastrophic forgetting (Khetarpal et al. 2022). It is therefore important to approach traffic control with Continual Learning (CL), an RL paradigm that takes non-stationarity into account.

CL agents need to be able to adapt quickly to changes in the environment (Khetarpal et al. 2022). DQN is an off-policy algorithm (Mnih et al. 2015), which have more variance and converge slower than on-policy methods (Sutton and Barto 2018). An on-policy algorithm such as PPO (Schulman et al. 2017) may therefore be more suited for traffic control problems.

1.1 Research Objectives

Research Question

Can Proximal Policy Optimization (PPO) and experience replay be combined for effective learning in a continual learning environment?

Understand behaviour of PPO agents in continual learning environments

Investigate whether experience replay can be used to mitigate problems such as catastrophic forgetting

Propose and evaluate possible improvements

1.2 Methodology

Introduce how these objectives will be achieved

1.3 Dissertation Structure

Overview of diffrerent chapters

2. State of the Art

This chapter presents background knowledge on RL and CL. Related works on CL approaches and the use of RL in urban traffic control are also reviewed.

2.1 Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm in which a so-called agent develops a behaviour by learning how to maximise a numerical reward through interacting with its environment (Sutton and Barto 2018). This differs from both supervised and unsupervised learning (Yang 2019). Supervised learning uses labelled data to model the data. Unsupervised learning uses unlabelled data to identify structures in the data. In contrast to this, RL aims to maximise reward rather than learn patterns and structures in data.

The interactions between an agent and its environment can be formalised as a Markov Decision Process (MDP) (Sutton and Barto 2018). An MDP is defined by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a set of rewards $\mathcal{R}$ and a transition probability function p. The diagram in Figure 2.1 shows the process of the interaction. At each time step t, the agent selects an action $A_t \in \mathcal{A}$ based on the current state $S_t \in \mathcal{S}$. According to the dynamics of the environment, represented by the p-function, the environment emits a reward $R_{t+1} \in \mathcal{R}$ and the next state S_{t+1} in response. The general definition of the p-function is given by Equation 2.1.

$$p(s', r|s, a) = Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\} \tag{2.1}$$

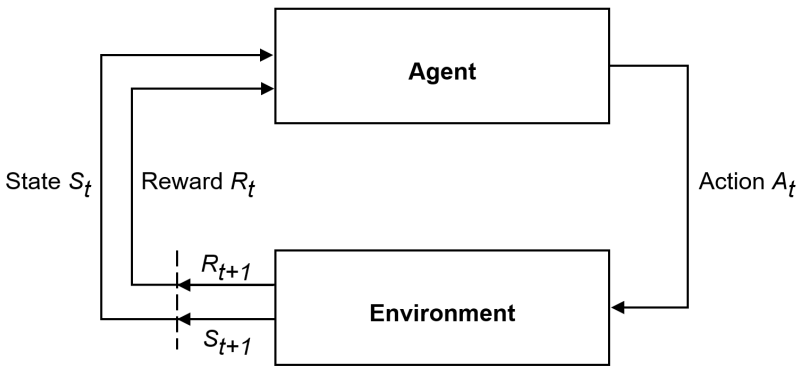

Figure 2.1: MDP framework (based on Sutton and Barto (2018))

This framework can be generalised further (as shown in Figure 2.2) to also describe Partially Observable MDPs (POMDP) (Silver 2015). POMDPs are cases in which the agent is unable to fully observe the state of the environment. In this framework, S_t^e denotes the environment

state. The observation O_t is the information about the environment state that is emitted to the agent. In a POMDP, the agent's internal representation of the state S_t^a differs from S_t^e and is constructed for example as a function of the previous state S_{t-1}^a and the observation O_t. An MDP is a special case where $O_t = S_t^e = S_t^a$.

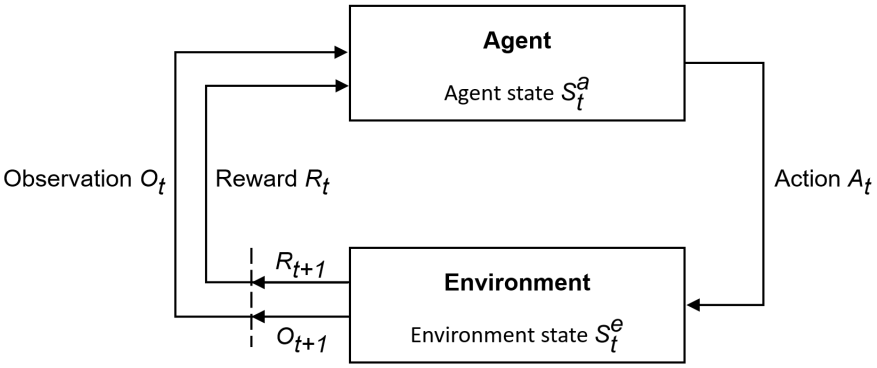

Figure 2.2: POMDP framework (based on Silver (2015))

2.1.1 Policies

The behaviour of an agent is defined by a policy π (Sutton and Barto 2018). The policy is defined by the probability distribution of actions given the state (Equation 2.2). In other words, the policy determines which action the agent executes at each time step.

$$\pi(a|s) = Pr\{A_t = a|S_t = s\} \tag{2.2}$$

During the training of an RL agent, there is a distinction between the behaviour policy and the target policy (Wang et al. 2022). The behaviour policy is the policy that the agent acts upon when generating training data. The target policy is the policy that is optimised through learning and should maximise the returns.

With this distinction, RL algorithms can be classified as either on-policy or off-policy (Wang et al. 2022). On-policy methods use the target policy as the behaviour policy. This means that in each round of optimisation, the target policy is used to generate the training samples. At the end of each round, these samples are discarded to avoid optimising on old policies. In off-policy methods on the other hand, the target and behaviour policies differ. The samples are generated by the behaviour policies and are stored in a buffer. Samples from this buffer are then used to update the target policy.

Due to the fact that samples in the buffer can be re-used, off-policy methods have a higher data efficiency compared to on-policy methods (Wang et al. 2022). However, off-policy methods have more variance and converge slower than on-policy methods (Sutton and Barto 2018).

2.1.2 Value Functions

The aim of an RL agent is to maximise the total reward, which is also referred to as the return G_t (Sutton and Barto 2018). This means that an agent should not only consider the immediate reward that is gained by taking an action, but future rewards should be considered as well. In RL, the discounted return (Equation 2.3) is usually considered. Here, the discount factor $\gamma \in [0, 1]$ weighs the importance of future rewards. If $\gamma = 0$, then the agent is only interested in the immediate rewards. As γ increases, so does the importance of future rewards and the agent becomes increasingly far-sighted.

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+1+k} \tag{2.3}$$

The expected return can be used to determine how good it is for an agent to be in a given state. This is quantified by the state-value-function $V(s)$, which gives the expected return when the agent starts in the state s (Equation 2.4). The higher the value, the better it is to be in the state. Using the Bellman equation, this can also be expressed in terms of the immediate reward and the discounted value of the next state (Equation 2.5).

$$V(s) = \mathbb{E}[G_t | S_t = s] \tag{2.4}$$

$$V(s) = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) | S_t = s] \tag{2.5}$$

Likewise, the expected return can be used to determine how good an action is in a given state. The action-value function $Q(s, a)$ gives the expected return when the agent starts in the state s and takes an action a (Equation 2.6). The higher the value, the better it is to take the action when in that state. This can also be transformed using the Bellman equation, which results in Equation 2.7.

$$Q(s, a) = \mathbb{E}[G_t | S_t = s, A_t = a] \tag{2.6}$$

$$Q(s, a) = \mathbb{E}[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{2.7}$$

An optimal value function $V_*(s)$ or $Q_*(s, a)$ is defined as the value function that is obtained when following an optimal policy π_* (Equations 2.8 and 2.9). The optimal policy is the policy that maximises the return for all states.

$$V_*(s) = \max_{\pi} V_\pi(s) \tag{2.8}$$

$$Q_*(s, a) = \max_{\pi} Q_\pi(s, a) \tag{2.9}$$

2.1.3 Reinforcement Learning Approaches

One distinction that can be made when categorising RL approaches is model-based vs. model-free (Sutton and Barto 2018). Model-based approaches require a model of the environment, which can be used to predict how the environment will behave. These models are used to evaluate potential next actions by considering all possible outcomes before they happen. This is referred to as planning.

Model-free approaches, on the other hand, learn through trial-and-error and do not require a model of the environment (Sutton and Barto 2018). Such approaches can be further classified as value-based methods, policy gradient methods or actor-critic methods.

Value-based algorithms aim to estimate the optimal value function (Wang et al. 2022). The optimal policy is implicitly given by the value function. For example, the target policy might be to behave greedily with respect to $Q_*(s, a)$. Examples for these algorithms are Q-Learning (Watkins and Dayan 1992) and Deep Q-Learning (DQN) (Mnih et al. 2015).

Policy gradient algorithms aim to optimise the target policy directly (Wang et al. 2022). Here, the parameterised policy $\pi_\theta(a|s) = Pr(a|s, \theta)$ is updated based on the gradient of a policy objective function $J(\theta)$. According to the policy gradient theorem, this can be calculated using Equation 2.10 (Silver 2015). The benefit of this approach is that the policy does not become deterministic and therefore continues to allow for exploration (Sutton and Barto 2018). An example of a policy gradient algorithm is REINFORCE (Williams 1992).

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[Q_{\pi_\theta}(s, a) \cdot \nabla_\theta log \pi_\theta(a|s)] \tag{2.10}$$

Actor-critic methods are methods that learn a policy and a value function at the same time (Wang et al. 2022). The actor generates policies, interacts with the environment and updates the target policy using the policy gradient. The critic learns the state-value function $V(s)$, action-value function $Q(s, a)$ or the advantage function $A(s, a)$ and uses this to evaluate the actor's policy. The advantage is defined in Equation 2.11 and represents the relative value of taking an action in a given state. This can be used instead of $Q(s, a)$ when calculating the gradient of the objective function (Equation 2.10) to reduce the variance of the gradients.

$$A(s, a) = Q(s, a) - V(s) \tag{2.11}$$

2.2 Continual Learning

Definition

2.2.1 Continual Learning Problems

Classification of problems

non-stationarity drivers etc.

2.2.2 Continual Learning Agents

ideal attributes of CL agent

CL Approaches

Explicit knowledge retention will be reviewed on more detail.

2.2.3 Explicit Knowledge Retention

Latent Parameter Storage

Definition

Discuss papers that do this

Distillation Based

Definition

Discuss papers that do this

Rehearsal Based

Definition

Discuss papers that do this

2.3 Reinforcement Learning in Urban Traffic Control

3. Design

3.1 Proposed Algorithm

Explain PPO and the approach to mitigate catastrophic forgetting

3.2 Urban Traffic Control Environment

Define environment

3.3 Experiments

Define the different experiments. Hypothesis?

4. Implementation

Tools used etc.

5. Evaluation

5.1 Experimental Results

Present results and discuss them

5.2 Future Work

Improvements that could be done in the future

6. Conclusion

Summary

References

Department of Transport (2023). *The Economic Cost of Congestion in the Greater Dublin Area 2022-2040*. Irish government.

Khetarpal, Khimya et al. (2022). "Towards Continual Reinforcement Learning: A Review and Perspectives". In: *Journal of Artificial Intelligence Research* 75. DOI: https://doi.org/10.48550/arXiv.2012.13490.

Liu, Junxiu et al. (2023). "Multiple intersections traffic signal control based on cooperative multi-agent reinforcement learning". In: *Information Sciences* 647, p. 119484. DOI: https://doi.org/10.1016/j.ins.2023.119484.

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518, pp. 529–533. DOI: doi:10.1038/nature14236.

Rasheed, Faizan et al. (2020). "Deep Reinforcement Learning for Traffic Signal Control: A Review". In: *IEEE Access* 8, pp. 208016–208044. DOI: 10.1109/ACCESS.2020.3034141.

Schulman, John et al. (2017). *Proximal Policy Optimization Algorithms*. DOI: https://doi.org/10.48550/arXiv.1707.06347.

Silver, David (2015). *Lectures on Reinforcement Learning*. URL: https://www.davidsilver.uk/teaching/.

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: an introduction*. 2nd ed. Cambridge, MA: The MIT Press. ISBN: 978-0-262-03924-6.

TomTom (2023). *Annual TomTom Traffic Index: The cost of driving has reached new highs around the world*. URL: https://www.tomtom.com/newsroom/press-releases/general/260960154/the-cost-of-driving-has-reached-new-highs-around-the-world/ (visited on 12/08/2023).

Wang, Xu et al. (2022). "Deep Reinforcement Learning: A Survey". In: *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15. DOI: 10.1109/TNNLS.2022.3207346.

Watkins, Christopher J. C. H. and Peter Dayan (1992). "Q-learning". In: *Machine Learning* 8.3, pp. 279–292. DOI: https://doi.org/10.1007/BF00992698.

Williams, Ronald J. (1992). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3, pp. 229–256. DOI: https://doi.org/10.1007/BF00992696.

World Health Organisation (2022). *Ambient (outdoor) air pollution*. URL: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health (visited on 12/08/2023).

Yang, Xin-She (2019). *Introduction to Algorithms for Data Mining and Machine Learning*. London: Academic Press. ISBN: 978-0-12-817216-2.