Traffic Light Control Using Deep Policy-Gradient and Value-Function Based Reinforcement Learning

April 2017 · IET Intelligent Transport Systems 11(7)
DOI:10.1049/iet-its.2017.0153

Authors:

Sajad Mousavi
Harvard Medical School

Michael Schukat
National University of Ireland, Galway

Peter Corcoran

Enda Howley
National University of Ireland, Galway

Abstract and Figures

Recent advances in combining deep neural network architectures with reinforcement learning techniques have shown promising potential results in solving complex control problems with high dimensional state and action spaces. Inspired by these successes, in this paper, we build two kinds of reinforcement learning algorithms: deep policy-gradient and value-function based agents which can predict the best possible traffic signal for a traffic intersection. At each time step, these adaptive traffic light control agents receive a snapshot of the current state of a graphical traffic simulator and produce control signals. The policy-gradient based agent maps its observation directly to the control signal, however the value-function based agent first estimates values for all legal control signals. The agent then selects the optimal control action with the highest value. Our methods show promising results in a traffic network simulated in the SUMO traffic simulator, without suffering from instability issues during the training process.

Deep reinforcement… | The intersection geometry for th… | A comparison of performance of… | Average Cumulative del… | Average queue length of the…

Figures - uploaded by Sajad Mousavi Author content

Public Full-text 1

Traffic Light Control Using Deep Policy-Gradient a Value-Function Based Reinforcement Learning

Seyed Sajad Mousavi
Discipline of IT
National University of Ireland, Galway
s.mousavi1@nuiglaway.ie

Michael Schukat
Discipline of IT
National University of Ireland, Galway
michael.schukat@nuigalway.ie

Enda Howley
Discipline of IT
National University of Irel Galway
enda.howley@nuigal

arXiv:1704.08883v2 [cs.LG] 27 May 2017

ABSTRACT

Recent advances in combining deep neural network architectures with reinforcement learning techniques have shown promising potential results in solving complex control problems with high dimensional state and action spaces. Inspired by these successes, in this paper, we build two kinds of reinforcement learning algorithms: deep policy-gradient and value-function based agents which can predict the best possible traffic signal for a traffic intersection. At each time step, these adaptive traffic light control agents receive a snapshot of the current state of a graphical traffic simulator and produce control signals. The policy-gradient based agent maps its observation directly to the control signal, however the value-function based agent first estimates values for all legal control signals. The agent then selects the optimal control action with the highest value. Our methods show promising results in a traffic network simulated in the SUMO traffic simulator, without suffering from instability issues during the training process.

CCS Concepts

•Theory of computation → Sequential decision making;

Keywords

Traffic control, Reinforcement learning, Deep learning, Policy gradient method, Value-function method, Artificial neural networks

1. INTRODUCTION

With regard to fast growing population around the world, the urban population in the 21st century is expected to increase dramatically. Hence, it is imperative that urban infrastructure is managed effectively to contend with this growth. One of the most critical consideration when designing modern cities is developing smart traffic management systems. The main goal of a traffic management system is reducing traffic congestion which nowadays is one of the major issues of megacities. Efficient urban traffic management results in timea and financial savings as well as reducing CO_2 emission into atmosphere. To address this issue, a lot of solutions have been proposed [23, 4, 1, 22]. They can be roughly classified into three groups. The first is pre-timed signal control, where a fixed time is determined for all green phases according to historical traffic demand, without considering possible fluctuations in traffic demand. The second is vehicle-actuated signal control where, traffi formation is used, provided by inductive loop an equipped intersection to decide to control e.g. extending or terminating a green phase. adaptive signal control, where the signal timi managed and updated automatically accordin rent state of the intersection (i.e. traffic de length of vehicles in each lane of the intersect fic flow fluctuation) [13]. In this study, we a in the third approach and aim to propose two ods for traffic signal control by leveraging rec in machine learning and artificial intelligence f

Reinforcement learning [34] as a machine l nique for traffic signal control problem has led results [4, 30] and has shown a promising pot It does not need to have a perfect knowledge ronment in advance, for example traffic flow. are able to gain knowledge and model the dyr environment just by interacting with it. A r learning agent learns based on trial and error. scalar reward after taking each action in the The obtained reward is based on how well the ta and the agent's goal is to learn an optimal con the discounted cumulative reward is maximized interaction with its environment. Aside from t reinforcement learning has been applied to a nu world problems such as cloud computing [12, 1

Typically the complexity of using reinforcer in real world applications such as traffic signal grows exponentially as state and action spaces deal with this problem, function approximatic and hierarchical reinforcement learning appro used. Recently, deep learning has gained huge a has been successfully combined with reinforcer techniques to deal with complex optimization p as playing Atari 2600 games [27], Computer [33], etc., where the classical RL methods coul optimal solutions. In this way, the current stat ronment is fed into a deep neural net (e.g. a neural network [20]) trained by reinforcement l niques to predict the next possible optimal act

Inspired by the successes of combining reinfor ing with deep learning paradigm and with regar plex nature of environment of traffic signal co in this paper we aim to use the effectiveness deep reinforcement learning to build adaptive methods in order to optimize the traffic flow. few previous studies have tried to apply deep r learning in the traffic signal control problem [38, 14], in this research the state representation is different. Also, One of our methods uses policy gradient method which does not suffer from oscillations and instabilities during training process and can take full advantage of the available data of the environment to develop the optimal control policy.

We propose adaptive signal controllers by combination two reinforcement learning approaches (i.e. policy gradient and action-value function) and a deep convolution neural network, which perceive embedded camera observations in order to produce control signals in an isolated intersection. We conduct simulated experiments with our proposed methods in SUMO traffic simulator.

The rest of the paper is organized as follows. Section 2 provides related work in the area of traffic light control (TLC). Section 3 gives a brief review of reinforcement learning techniques which we have used in this research. Section 4 presents how to formulate the TLC problem as a reinforcement learning task and the proposed methods to solve the task. Then Section 5 provides simulation results and the performance of the proposed approaches. Finally Section 6 concludes the paper and give some directions for future research.

2. RELATED WORK

A lot of research has been done in academic and industry communities to build adaptive traffic signal control systems. In particular, significant research has been conducted employing reinforcement learning methods in the area of traffic light signal control [39, 2, 7]. These works have achieved promising results. However, their simulation testbeds have not been mature enough to be comparable with more realistic situations. Developing advance traffic simulation tools have made researchers develop novel state representation and reward functions for reinforcement learning algorithms, which could consider more aspects of complexity and reality of real-world traffic problems [13, 1, 8, 3]. All this these attempts viewed the traffic light control problem as a fully observable Markov decision process (MDP) and investigated whether Q-learning algorithm can be applied to it. However, Richter's study formulated the traffic problem as a partially observable MDP (POMDP) and applied policy gradient methods to guarantee local convergence under a partial observable environment [31].

By utilizing advances in deep learning and its application to different domains [10, 11], deep learning has gained attention in the area of traffic management systems. Previous research has used deep stacked autoencoders (SAE) neural networks to estimate Q-values, where each Q-value is corresponding to each available signal phase [21]. It considered measures of speed and queueing length as its state in each time step of learning process of its proposed method. Two recent studies by [38, 14] provided deep reinforcement learning agents that used deep Q-netwok [27] to map from given states to Q-values. Their state representations were a binary matrix of the positions of vehicles on the lanes of an intersection, and a combination of the presence matrix of vehicles, speed and the current traffic signal phase, respectively. However, we use raw visual input data of the traffic simulator snapshots as system states. Moreover, in addition to estimating Q-function, one of the proposed methods directly maps from the input state to a probability distribution over actions (i.e. signal phases) via deep policy gradient method.

3. BACKGROUND

In this section, we will review Reinforceme (RL) approaches and briefly describe how RL real world problems where the number of state are extremely high so that the regular reinforcer techniques cannot deal with them.

3.1 Reinforcement Learning

A common reinforcement learning [34] settin Figure 1 where an RL agent interacts with an The interaction is continued until reaching a t or the agent meets a termination condition. problems that RL techniques are applied to, a Markov decision processes (MDPs). A MDP is five-tuple $< S, A, T, R, \gamma >$, where S is the se the state space of the environment, A is the set the action space that the agent can use in ord with the environment, T is the transition funct the probability of moving between the enviror R is the reward function and $\gamma \in [0, 1]$ is know count factor, which models the importance of tl immediate rewards. At each time step t, the ag the state $s_t \in S$ and, based on its observatio action a_t. Taking the action, leads to the sta vironment transitions to the next states $s_{t+1} \in$ the transition function T. Then, the agent rec r_t which is determined by the reward function

The goal of the learning agent in reinforcem framework is to learn an optimal policy $\pi : S$ which defines the probability of selecting actic s_t, so that with following the underlying policy cumulative discounted reward over time is ma: discounted future reward, R_t at time t is defin

$$R_t = E[\sum_{k=0}^{\infty} \gamma^k r_{t+k}],$$

where the role of the discount factor γ is to worth of immediate and future rewards. In mo problems, there are many states and actions it impossible to apply classic reinforcement le niques, which consider tabular representations f and action spaces. For example, in the prol fic light optimization, that we interest in thi state space is continuous. Hence, it is common tion approximators [42] or decomposition and techniques like Hierarchical Reinforcement Lea approaches [5, 15, 28] and advance HRL [16].

Different forms of function approximators with reinforcement learning techniques. For e ear function approximation, a linear combinati of state and action spaces f and learned we $\sum_i f_i w$) or a non-linear function approximatio ral network). Until recently, the majority of forcement learning has been applying linear fun imatiors. More recently, deep neural networks as convolutional neural networks (CNNs), recu networks (RNNs), stacked auto-encoders (SAE also been commonly used as function approxima reinforcement learning tasks [19, 26]. The inter are referred to [29] for a review of using deep neu

Figure 1: Deep reinforcement learning agent of traffic signal control.

with reinforcement learning framework.

3.2 Deep learning and Deep Q-learning

Deep learning techniques are one of the best solutions to address high dimensional data and extract discriminative information from the data. Deep learning algorithms have the capability of automating feature extraction (the extraction of representations) from the data. The representation are learnt through the data which are fed directly into deep nets without using human knowledge (i.e. automated feature extraction). Deep learning models contain multiple layers of representations. Indeed, it is a stack of building blocks such as auto-encoders, Restricted Boltzmann Machines (RBMs) and convolutional layers. During training, the raw data is fed into a network consisting of multiple layers. The output of the each layer which is nonlinear feature transformations, is used as inputs to the next layers of the deep neural network. The output representation of the final layer can be used for constricting classifiers or those applications which can have the better efficiency and performance with abstract representation of the data in a hierarchical manner as inputs. A nonlinear transformation is applied at each layer on its input to try to learn and extract underlying explanatory factors. Consequently, this process learns a hierarchy of abstract representations.

One of the main advantages of deep neural networks is the capability of automating feature extraction from row input data. A deep Q-learning Network (DQN) [26] uses this benefit of deep learning in order to represent the agent's observation as an abstract representation in learning an optimal control policy. The DQN method aggregates a deep neural network function approximator with Q-learning to learn action value function and as a result a policy π, the behaviour of the agent which tells the agent what action should be selected for each input state. Applying non-linear function approximators such as neural networks with model-free reinforcement learning algorithms in high-dimensional continuous state and action spaces, has some convergence problems [37]. The reasons for these issues are: 1) Consecutive states in reinforcement learning tasks have correlation. 2) The underlying policy of the agent is changing frequently, because of slight changes in Q-values. To cope with these problems, the DQN provides some solutions which improve the performance of the algorithm significantly. For the prob-

lem of correlated states, DQN uses the previou
experience replay approach [24]. In this way,
step, the DQN stores the agent's experience (s
into a date set D, where s_t, a_t, and r_t are th
sen action and received reward, respectively ar
state at the next time step. To update the
DQN utilizes stochastic minibatch updates wi
random sampling from the experience replay r
vious observed transitions) at training time.
strong correlations between consecutive samp
approach to deal with aforementioned conver
which we also examine in this research, is the
ent methods. This approach has demonstrate
vergence properties in some RL problems [35].

3.3 Policy Gradient Methods

A Policy Gradient (PG) method tries to op
rameterized policy function by gradient descent
deed, policy gradient methods are interested
policy space to learn policies directly, instead
ing state-value or action-value functions. Un
ditional reinforcement learning algorithms, PG
not suffer from the convergence problems of esti
functions under nonlinear function approximat
environments which might be partially obser
They can also deal with the complexity of con
and action spaces better than purely value-ba
[35]. Policy gradient methods estimate policy
ing Monte Carlo estimates of the policy gradier
methods are guaranteed to converge to a loca
their parametrized policy function. However,
methods result in high variance in their gradie
Hence, in order to reduce the variance of the g
mators, some methods subtract a base line func
policy gradients. The baseline function can be
different manners [32, 40]. By inspiring these fe
methods and successes of neural networks in a
ture abstractions, we use deep neural networks
an optimal traffic control policy directly in the
control problem.

4. SYSTEM DESCRIPTION

In this section, we will formulate traffic light
lem as a reinforcement learning task by describi
actions and reward function. We then present
a deep neural network and how to train the ne

4.1 State Representation

We represent the state of the system as an in
or a snapshot of the current state of a graphi
(e.g. SUMO-GUI [18]) which is a vector of row
of current view of the intersection at each ste
tion (as shown in Figure 1). This kind of re
is like putting a camera on an intersection w
it to view the whole intersection. The state re
in the traffic light control literature usually u
representing the presence of a vehicle at the in
Boolean-valued vector where a value 1 indicates
of a vehicle and a value 0 indicates the absen
cle [38, 36], or a combination of the presence
another vector indicating the vehicle's speed
intersection [14]. Regardless of these states rep
that are using a prior knowledge provided, th

sumptions which are not generalizable for the real world. For instance, they discretize a lane segment of an intersection into cells with a constant length c which is supposed to be the vehicle length to build the vehicle's speed and presence vectors. However, by feeding the state as an image to a convolutional neural network, the system can detect the location and presence of all vehicles with different lengths and as result the vehicles' queue on each lane. Furturmore, by stacking a history of consecutive observations as input, the convolutional layers of a deep network are able to estimate velocity and travel direction of vehicles. Hence, the system can implicitly benefit from these information as well.

4.2 Action Set

To control traffic signal phases, we define a set of possible actions $A = \{\text{North/South Green (NSG), East/West Green (EWG)}\}$. NSG allows vehicles to pass from North to South and vice versa, and also indicates the vehicles on East/West route should stop and not proceed through the intersection. EWG allows vehicles to pass from East to West and vice versa, and implies the vehicles on North/South route should stop and not proceed through the intersection. At each time step t, an agent regarding its strategy chooses an action $a_t \in A$. Depending the selected action, the vehicles on each lane are allowed to cross the intersection.

4.3 Reward Function

Typically an immediate reward $r_t \in \mathbb{R}$ is a scalar value which the agent receives after taking the chosen action in the environment at each time step. We set the reward as the difference between the total cumulative delays of two consecutive actions, i.e.

$$r_t = D_{t-1} - D_t, \tag{2}$$

where D_t and D_{t-1} are the total cumulative delays in the current and previous time steps. The total cumulative delay at time t, is the summation of the cumulative delay of all the vehicles appeared from $t = 0$ to current time step t in the system. The positive reward values imply the taken actions led to decrease the total cumulative delay and the negative rewards imply an increase in the delay. With regard to the reward values, the agent may decide to change its policy in certain states of the system in the future.

4.4 Agent's Policy

The agent chooses the actions based on a policy π. In the policy-based algorithm, the policy is defined as a mapping from the input state to a probability distribution over actions A. We use the deep neural network as the function approximator and refer its parameters θ as policy parameters. The policy distribution $\pi(a_t|s_t;\theta)$ is learned by performing gradient descent on the policy parameters. In the value-function based algorithm, the deep neural network is utilized to estimate the action-value function. The action-value function maps the input state to action values, which each represents the future reward that can be achieved for the given state and action. The optimal policy can then be extracted by performing a greedy approach to select the best possible action.

4.5 Objective Function and System Training

There are many measures such as maximizing throughput, minimizing and balancing queue length, minimizing the

Algorithm 1 Deep Value-Function based r
learning agent of traffic signal control with e
play

Initialize parameters, θ with random value
Initialize replay memory M with capacity
for each simulation **do**
  initialize s with current view of the inte
  repeat # each step in th
    choose action a according to ϵ-greedy
    take action a, observe reward r and ne
    store transition (s, a, r, s') in M
    $s \leftarrow s'$
    $b \leftarrow$ sample random minbatch of tra
    the replay memory, M
    for each transition (s_j, a_j, r_j, s'_j) in b
      if s'_j is terminal **then**
        $y_i \leftarrow r_j$
      else
        $y_j = r_j + \gamma max_{a'} Q({s'}_j, a'; {\theta^-}_{i-1})$
      end if
      update parameters θ according to e
    end for
  until s is terminal
end for

delay, etc. in the traffic signal management consider as the learning agent's objective func research, the agent aims to maximize the red total cumulative delay, which empirically has to maximize throughput and to reduce queue details discussed in Section 5.3).

The objective of agent is to maximize the exp lative discounted reward. We aim to maximiz under the probability distribution $\pi(a_t|s_t;\theta)$:

$$J(\theta) = E_{\pi_\theta}[\sum_{t=0}^{T} \gamma^t r_t] = E_{\pi_\theta}[R].$$

We divide the system training based two RL Value-function based and Policy-based. in **val based approach**, the value function, $Q_\pi(s, a)$ follows:

$$Q_\pi(s, a) = E_\pi[r_t + \gamma max_{a'} Q(s', a')|s$$

Where it is implicit that $s, s' \in S$ and $a \in$ function can be parameterized, $Q(s, s; \theta)$ wit vector θ. Typically, the gradient-descent meth to learn parameters, θ by trying to minimize loss function of mean-squared error in Q value

$$J(\theta) = E_\pi[(r + \gamma max_{a'} Q(s', a'; \theta) - Q(s, a$$

Where $r + \gamma max_{a'} Q(s', a'; \theta)$ is the target v DQN algorithm, a target Q-network is used to instability problem of the policy. The netwo with the target Q-network to obtain consiste targets by keeping the weight parameters (θ^-) u learning target fixed and updating them perio N steps through the parameters of the main net target value of the DQN is represented as follo

$$y_i = r + \gamma max_{a'} Q(s', a'; {\theta^-}_{i-1})$$

Figure 2: The intersection geometry for the traffic simulation.

Where θ^- is parameters of the target network. The stochastic gradient descent method is used in order to optimize equation (5). The parameters of the deep Q-learning algorithm are updated as follows:

$$\theta_i \leftarrow \theta_{i-1} + \alpha(y_i - Q(s, a; \theta_i)) \bigtriangledown_{\theta_i} Q(s, a; \theta_i) \qquad (7)$$

Where y_i is the target value for iteration i and α is a scalar learning rate. Algorithm 4.4 presents the pseudo-code for the training algorithm.

In **policy-based approach**, The gradient of the objective function represented in equation (3) is given by:

$$\bigtriangledown_\theta J = \sum_{t=0}^{T} E_{\pi_\theta}[\bigtriangledown_\theta log(a_t|s_t;\theta)R_t]. \qquad (8)$$

This equation (8) is standard learning rule of the REINFORCE algorithm [41]. It updates the policy parameters θ in the direction $\bigtriangledown_\theta log(a_t|s_t;\theta)$ so that the probability of action a_t at state s_t is increased if it has led to high cumulative reward, however it is decreased if the action has result in a low reward. The gradient estimate in equation 2 results to have high variance. It is common to reduce the variance by subtracting a baseline function $b_t(s_t)$ from the return R_t, without changing expectation. Commonly an estimate of the state value function is used as the baseline, $b_t(s_t) = V^{\pi_{\theta_v}}(s_t)$. Thus, the adjusted gradient is $\bigtriangledown_\theta log(a_t|s_t;\theta)(R_t - b_t(s_t))$. The value $R_t - b_t$ is known as the *advantage function*.

With regard to the advantage actor-critic method [25], computing a single update is done by selecting actions using the underlying policy for up to M steps or till a terminal state is met. In this way, the agent obtains up to M rewards from the environment at each update point and updates the policy parameters after every $n \leq M$ steps regarding n-step returns. The vector parameters θ is updated through the stochastic gradient descent method:

$$\theta \leftarrow \theta + \alpha \sum_t \bigtriangledown_\theta log(a_t|s_t;\theta) A(s_t, a_t; \theta, \theta_v), \qquad (9)$$

where $A(s_t, a_t; \theta, \theta_v)$ is an estimate of the *advantage function* corresponding $\sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n V(s_{t+n}; \theta) - V(s_t; \theta_v)$, where n might have different values with respect to the state, up to M. this process is an actor-critic algorithm, the policy $\pi(a_t|s_t;\theta)$ refers to the actor and the estimate value function $V^{\pi_{\theta_v}}(s_t)$ implies to the critic [rithm 4.5 shows the pseudo-code for the trainin

Algorithm 2 Deep Policy-Gradient based r learning agent of traffic signal control

Initialize parameters, θ, θ_v with random va
Initialize step counter $t \leftarrow 0$
for each simulation **do**
  initialize s with current view of the inter
  $t_{start} = t$
  repeat
    perform action a according to policy π
    observe reward r and next state s'
    $t \leftarrow t + 1$
  until s is terminal **or** $t - t_{start} == M$ (
  if s is terminal **then**
    $R = 0$
  else
    $R = V(s; \theta_v)$
  end if
  for $i \in \{t-1, ..., t_{start}\}$ **do** $n <= M$ times
    $R \leftarrow r_i + \gamma R$
    $\theta \leftarrow \theta + \alpha \bigtriangledown_\theta log(a_i|s_i;\theta)(R - V(s_i;\theta$
    $\theta_v \leftarrow \theta_v + \frac{\partial(R - V(s_i;\theta_v))^2}{\partial\theta_v}$
  end for
end for

5. EXPERIMENT AND RESULTS

In this section, we present the simulation where our experiments have been done. We t the details of the deep neural network utilis hyper-parameters to represent the agent's poli

5.1 Experiment Setup

We have used the Simulation of Urban MObi [18] tool to simulate traffic in all experiment a well-known open source traffic simulator wl useful Application Programming Interfaces (Graphical User Interface (GUI) view to mod networks as well as some possibilities to handle t ticular, we utilised SUMO-GUI v0.28.0. as it a snapshots of each step of the simulation. The geometry used in this study is shown in Figure 4 incoming lanes to the intersection and four o from the intersection. To generate traffic dema ferent directions (i.e. north-to-south and wes vice versa) to the road network, a uniform pro tribution with the probability 0.1 was used.

5.2 System Architecture and Hyper

We took the snapshots from the SUMO-GUI basic pre-processing. The snapshots are conver green-blue (RGB) representation to gray-scale them to 128 × 128 frames. To enable our syste rize a history of the past observations, we sta four frames of the history and provided them t as input. So, the input to the network was a 1 image. We applied approximately the same an the Deep Q-Network (DQN) algorithm introdu

Table 1: Comparison of performance of the proposed methods against the SNN metho

Evaluation Metric (μ, σ; $n = 100$)	Policy-based	Value-function based	SNN
Queue (Vehicles)	(1.79, 0.073)	(1.74, 0.10)	(5.55, 0.73)
Cumulative Delay (s)	(11.25, 0.39)	(11.01, 0.69)	(41.40, 7.31)
Reward (r)	(6.14, 0.29)	(7.14, 0.44)	(1.73, 0.62)

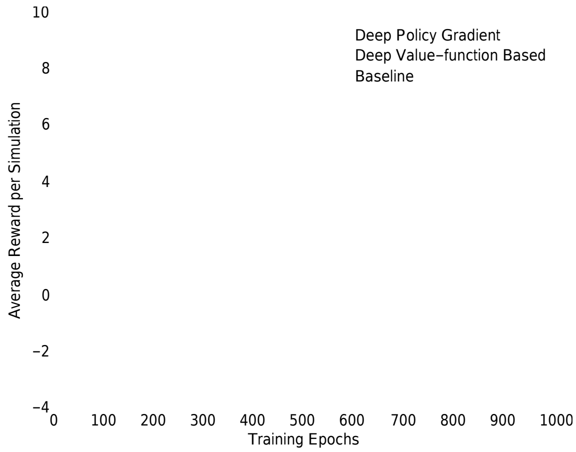

Figure 3: A comparison of performance of the average reward received during the evaluation time for the proposed method and the baseline.

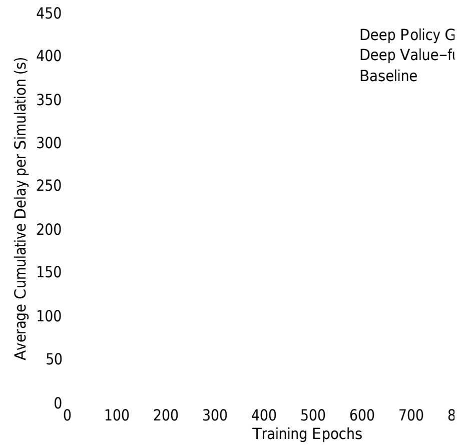

Figure 4: Average Cumulative delay per ing the suggested model and the baseline evaluation time.

et al. [26, 27]. The network consists a stack of two convolutional layers with filters 16 8×8 and 32 4×4 and strides 4 and 2, respectively. The final hidden layer is fully-connected with 256 hidden node. All three hidden layers are followed by a rectifier nonlinearity. The main difference with the network architecture of the DQN method is the last layer, where the last layer of DQN is a fully-connected linear layer with a number of output neurons (i.e. Q-values $Q(a, s)$) corresponding to each action in a given Atari 2600 game, while in policy-based model the last layer represents two set of outputs, a softmax output resulting in a probability distribution over the actions A (i.e. the policy $\pi(a, s)$), and a single linear output node resulting in the estimate of the state value function $V(s)$. For value-function model we used the architecture, the same as the DQN. The output layer is corresponding to action values. In all of our experiments, the discount factor was set to $\gamma = 0.99$ and all weights of the network were updated by the Adam optimizer [17] with a learning rate $\alpha = 0.00001$ and with mini batches of size M (up to 32), the maximum number of steps that the agent can take to follow its policy and afterwards needs to update it. The network was trained for about 1050 epoch, approximately 2 million time steps. Each epoch is corresponded 10 episodes and each episode was a complete SUMO-GUI simulation. The learned policies by the agent was evaluated every 10 episodes by running SUMO-GUI for 5 episodes and averaging the resulting rewards, total cumulative delay and queue length.

To evaluate our proposed method we also built a shallow neural network (SNN) with one hidden layer. The hidden layer has 64 hidden nodes followed by a rectifier nonlinearity. The output layer is a fully-connected linear layer with a number of output neurons corresponding tc signal phase in the intersection. Two vectors input state of the network. The first represent ber of queued vehicles at the lanes of the inte North, South, East and West) and the second c to the current traffic signal phase of the inters is trained with the same hyper-parameters and method (i.e. the gradient decent algorithm) as methods.

5.3 Results and Discussion

To evaluate the performance of the proposed compared them against a baseline traffic cont troller that gives an equal fixed time to each intersection. We ran SUMO-GUI simulator for model using the configuration setting explaine 5.2 and compared the average reward, average lative delay and average queue length achieved line. Figure 3 shows the received average rewa agent follows a certain policy. As shown in Figu posed method performs significantly better thar and results more reward magnitudes by doing This gradually increasing reward reflects the ity to learn an optimal control policy in a sta Unlike using deep reinforcement learning for es Q-values in traffic light optimisation problem [posed agent doesn't suffer stability issues. In or the learned policy by the agent, two of the m performance metrics in the traffic signal contro implemented: the cumulative delay and queue ures 4 and 5 illustrate the performance comp leaning agent regarding average cumulative de

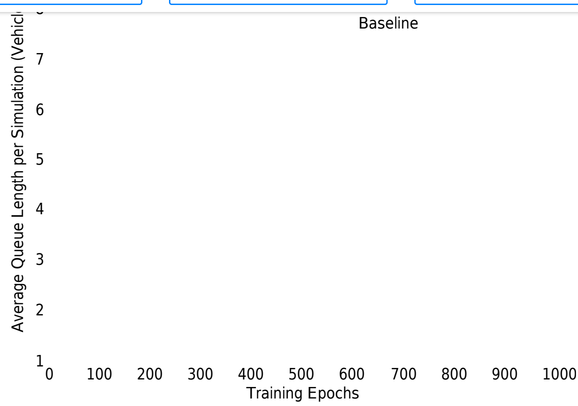

Figure 5: Average queue length of the intersection using the proposed model and the baseline during the evaluation time.

average queue length metrics, respectively, to the baseline, while the agent is following the learning policy over time. The plots clearly show the agent is able to find a policy resulting minimizing queue length and total cumulative delay. Moreover, these graphs reveal that by using the reward function for reducing cumulative delay, the intersection queue length is reduced as well as the total cumulative delay of all vehicles.

We also compared the proposed methods with the SNN, which is a shallow neural network with one hidden layer. Table 1 reports a comparison of the proposed models and the SNN model in terms of the average and standard deviation (μ, σ) of average queue length, average cumulative delay time and the received average reward metrics. The results on Table 1 are calculated from the last 100 training epochs of each method. Comparing the metrics shown in Table 1, demonstrates that the proposed models significantly outperform the SNN method. Based on the data in Table 1 we can induce 67% and 72% reductions in the average cumulative delay and queue length for the policy gradient method and 68% and 73% reductions for value-function based method compared to the SNN. Furthermore, we can see that the proposed methods have received average rewards superior to the SNN. Considering these results, it is obvious that the policy gradient and value-function agents could learn the control policies better than the SNN approach.

6. CONCLUSION

In this paper, we applied deep reinforcement learning algorithms with focusing on both policy and value-function based methods to traffic signal control problem in order to find optimal control policies of signalling, just by using raw visual input data of the traffic simulator snapshots. Our approaches have led to promising results and showed they could find more stable control policies compared to previous work of using deep reinforcement learning in traffic light optimization. In our work, we developed and tested the proposed methods in a small application, extending the work for more complex traffic simulations, for instance consider-

coordination problem between agents would b
for future research.

Acknowledgments

We would like to thank FotoNation company f
to work with their GPU cluster.

REFERENCES

[1] M. Abdoos, N. Mozayani, and A. L. Bazza
multi-agent system for traffic signals cont
Engineering Applications of Artificial Inte
26(5):1575–1587, 2013.

[2] B. Abdulhai, R. Pringle, and G. J. Karako
Reinforcement learning for true adaptive
control. *Journal of Transportation Engine*
129(3):278–285, 2003.

[3] I. Arel, C. Liu, T. Urbanik, and A. Kohls.
Reinforcement learning-based multi-agent
network traffic signal control. *IET Intellig*
Transport Systems, 4(2):128–135, 2010.

[4] P. Balaji, X. German, and D. Srinivasan.
signal control using reinforcement learnin
IET Intelligent Transport Systems, 4(3):17

[5] A. G. Barto and S. Mahadevan. Recent ac
hierarchical reinforcement learning. *Discr*
Dynamic Systems, 13(4):341–379, 2003.

[6] J. Baxter, P. L. Bartlett, and L. Weaver.
with infinite-horizon, policy-gradient estin
Journal of Artificial Intelligence Research
2001.

[7] E. Brockfeld, R. Barlovic, A. Schadschneid
M. Schreckenberg. Optimizing traffic light
cellular automaton model for city traffic.
Review E, 64(5):056132, 2001.

[8] Y. K. Chin, N. Bolong, A. Kiring, S. S. Ya
K. T. K. Teo. Q-learning based traffic opt
management of signal timing plan. *Intern*
Journal of Simulation, Systems, Science &
Technology, 12(3):29–35, 2011.

[9] T. Degris, P. M. Pilarski, and R. S. Sutton
reinforcement learning with continuous ac
practice. In *American Control Conference*
2012, pages 2177–2182. IEEE, 2012.

[10] L. Deng. A tutorial survey of architecture
algorithms, and applications for deep lear
APSIPA Transactions on Signal and Info
Processing, 3:e2, 2014.

[11] M. Duggan, J. Duggan, E. Howley, and E.
autonomous network aware vm migration
cloud data centres. In *Cloud and Autonom*
Computing (ICCAC), 2016 International
on, pages 24–32. IEEE, 2016.

[12] M. Duggan, K. Flesk, J. Duggan, E. Howl
E. Barrett. A reinforcement learning appr
dynamic selection of virtual machines in c
centres. In *Sixth International Conference*
Innovating Computing Technology. IEEE,

[13] S. El-Tantawy, B. Abdulhai, and H. Abde
Multiagent reinforcement learning for inte

Citations (133) References (53)

... The single objective RL is shown in Fig. 4. The learning agent optimizes the parameter that is used to design the reward function. In traffic signal control, the parameter can be the traffic feature of road intersection like queue length [12], waiting time [11,38], delay [14,39], etc. Here delay is . ...

... The state is designed in many ways in the literature. Some approaches take single traffic feature information for state [22,48] and some consider multiple features [14,38,39,49]. To give the agent more information about the environment, in CRA the state is designed with three traffic features information. ...

... Delay as a reward function is designed using Equation 6. This reward function is used in multiple studies [14,39] since it is also a crucial parameter for minimize. 4. HRA: This approach is proposed in [19]. ...

Adaptive traffic signal control system using composite reward architecture based deep reinforcement learning

Article Full-text available

Jan 2021 · IET INTELL TRANSP SY

Abu Rafe Md. Jamil · Kishan Kumar Ganguly · Naushin Nower

View

... En effet, le peu d'usagers autorisé à sortir serait contraint d'attendre inutilement à certains feux de signalisation, bien que la chaussée soit complètement dégagée. Les méthodes de résolution en ligne doivent donc prendre des décisions à la volée, en tenant compte de la situation présente et en émettant parfois des hypothèses sur le futur (par exemple en analysant un flux de données de capteurs ou de vidéo-surveillance au cours de l'optimisation [114]). ...

... D'autre part, les méthodes en ligne considèrent l'optimisation dynamique des réglages en tenant compte de l'information sur le trafic en temps réel (par exemple à l'aide d'un flux de vidéo-surveillance). Bien que ces méthodes en ligne s'avèrent efficaces à petite échelle, elles sont généralement difficiles à transposer à l'échelle d'une ville entière [160,114,54]. En outre, la grande majorité des feux de circulation est encore paramétrée par des réglages fixes [9,55,121,132,137,106]. Aussi, les contributions présentées dans la section 2.4.1 ainsi qu'aux chapitres 3, 4 et 5 se concentrent essentiellement sur des problèmes d'optimisation hors ligne pour la recherche du réglage optimal des feux de signalisation préprogrammés, selon la vraisemblance du trafic urbain dans les villes étudiées. ...

Méta-modélisation, simulation et optimisation de flux urbains

Thesis

Nov 2020

Florian Leprêtre

... Recent works Gao et al. (2017); Wan and Hwang (2018); Mousavi et al. (2017) use neural networks as function approximators to avoid the dimensionality and computing limitations of table based methods in large state-action spaces, showing DRL TSC can be more efficient than some earlier methods. The first two use discreet cell encoding vectors to represent the system, which are passed to a Convolutional Neural Network (CNN), whereas the second directly uses pixels in the same manner. ...

Reinforcement Learning for Traffic Signal Control: Comparison with Commercial Systems

Preprint Full-text available

Apr 2021

... During the decision process, the policy that the agent takes combines both exploitation of already learned policies and exploration of new policies that never met before. Studies using similar RL frameworks to manipulate ATSC are not rare in the last two decades and have provided beneficial references for research [14,15,16,17,18]. For instance, a single-agent model-free Q-learning algorithm was developed for optimizing signal timing in a single intersection [14].In this study, the authors used queue length as the state representation and accumulative delay between two action cycles as the reward. ...

Network-wide traffic signal control optimization using a multi-agent deep reinforcement learning

Preprint Full-text available

Apr 2021

... A widely used class of algorithms in the literature are value-based methods [41,43,45]. These algorithms try to extract the near optimal policy based on the value function, which is defined in Eq. 6. ...

Availability-aware and energy-aware dynamic SFC placement using reinforcement learning

Article Full-text available

Apr 2021 · J Supercomput

Guto Leoni · Judith Kelner · Patricia Takako Endo

... Recent advances in artificial intelligence (AI) and machine learning have made image-based modeling and analysis (e.g., classification, real time prediction, and image segmentation) even more successful in different applications [23,24,25,26]. Also, with the advent of nanotechnology semiconductors, a new generation of Tensor Processing Units (TPUs) and Graphical Processing Units (GPUs) can provide an extraordinary computation capability for data-driven methods [27]. ...

Aerial imagery pile burn detection using deep learning: The FLAME dataset

Article Full-text available

Mar 2021 · COMPUT NETW

Alireza Shamsoshoara · Fatemeh Afghah · Abolfazl Razi · Erik Blasch

... In 2015, deep reinforced learning was firstly introduced to traffic signal control optimization in [25] and further refined in 2016 by Van der Pol et al. [26], while considering the coordination of multiple intersections in a small network. In 2017, a traffic signal control policy has been trained by deep policy gradient and applied to a large traffic network by assuming multiple intersections could be controlled with the same agent [27], [28]. The result showed promising potential for policy-based reinforcement learning for traffic signal control. ...

Boosted Genetic Algorithm using Machine Learning for traffic control optimization

Preprint

Mar 2021

Tuo Mao · Mihaita Adriana Simona · Fang Chen · Hai L. Vu

... In [29] builds two kinds of reinforcement learning algorithms, namely, deep policy-gradient (PG) and value-function-based agents, that can predict the best possible traffic signal for traffic intersections. The adaptive traffic light control agent receives a snapshot of the current graphical traffic simulator and generates a control signal. ...

Recent development of smart traffic lights

Article Full-text available

Mar 2021

A'isya Nur Aulia Yusuf · Ajib Setyo Arifin · F.Y. Zulkifli

... The former aims to find suitable fixed-time signal plans based on historical traffic demand, while the latter dynamically adjusts the signal state according to the traffic information detected in real time. Although traffic responsive methods are technically sound, their performance depends heavily on real-time sensor systems [7] and they are generally difficult to apply to the whole city owing to the high operational cost [8,9]. Besides, the majority of traffic lights in real-world work under fixed signal timing plans and the traffic flows tend to repeat similar patterns like morning and evening peaks. ...

Surrogate-assisted cooperative signal optimization for large-scale traffic networks

Preprint

Mar 2021

Yongsheng Liang · Zhigang Ren · Lin Wang · Wenhao Du

... To solve this problem, a combination of RL with deep neural networks has shown promising results. The authors of the study performed in [36] used value function-based agents and deep policygradient to predict the optimal signal at intersections. A snapshot of the current state is received to these adaptive traffic light controls to produce signals at each time step. ...

Traffic Flow Management of Autonomous Vehicles Using Deep Reinforcement Learning and Smart Rerouting

Article Full-text available

Mar 2021

Anum Mushtaq · Irfan Ul Haq · Muhammad Usman · Omair Shafiq

Recommendations

Project

Predictive Communication for UAV Networks

Abolfazl Razi · Fatemeh Afghah · Jonathan Ashdown · [...] · Kurt Turck

Project

Intensive Care systems

Joerg Kampmann · Michael Schukat · E. Schwarzer · [...] · G. Lau

Project

Digital Certificate-based Port Knocking for Connected Embedded Systems

Basim Mahbooba · Michael Schukat

This research aims to reinforce existing port knocking methods with a digital certificate for alternative authentication among IoT devices. Such concepts will be complementary to other cryptographi ... [more]

Conference Paper Full-text available

Traffic Light Control Using Deep Reinforcement Learning Agent

April 2017

Sajad Mousavi · Michael Schukat · Enda Howley

Recent advances in combination deep neural network architectures and reinforcement learning techniques have shown promising

Thesis Full-text available

Researching Advanced Deep Learning Methodologies in Combination with Reinforcement Learning Techniqu...

December 2018

Sajad Mousavi · Michael Schukat · Enda Howley

Artificial intelligence (AI) field concerns to build autonomous agents that learn to do tasks successfully in complex and uncertain environments. AI provides powerful techniques which are used to solve many real-world problems ranging from computer science, industry, games, music to hospitals and medicine. What makes it applicable in various domains is a machine learning approach, which is the ... [Show full abstract]

Article

Deep Reinforcement Learning for Traffic Light Control in Vehicular Networks

March 2018

Xiaoyuan Liang · Xunsheng Du · Guiling Wang · Zhu Han

Existing inefficient traffic light control causes numerous problems, such as long delay and waste of energy. To improve efficiency, taking real-time traffic information as an input and dynamically adjusting the traffic light duration accordingly is a must. In terms of how to dynamically adjust traffic signals' duration, existing works either split the traffic signal into equal duration or extract ... [Show full abstract]

Conference Paper Full-text available

Deep Learning Methodologies in Combination with Reinforcement Learning Techniques

April 2016

Sajad Mousavi · Michael Schukat · Enda Howley

Before a reinforcement learning agent (software or hardware) can choose an action, it must have a good representation of the environment in which the agent is to be learned. Hence, perception is one of the key problems that must be solved before the agent can decide to select an optimal action to take. Learning good representations of high-dimensional state or action spaces is a major challenge ... [Show full abstract]

Article Full-text available

Using a Deep Reinforcement Learning Agent for Traffic Signal Control

November 2016

Wade Genders · Saiedeh N Razavi

Ensuring transportation systems are efficient is a priority for modern society. Technological advances have made it possible for transportation systems to collect large volumes of varied data on an unprecedented scale. We propose a traffic signal control system which takes advantage of this new, high quality data, with minimal abstraction compared to other proposed systems. We apply modern deep ... [Show full abstract]