Traffic Simulation & Navigation Environment in RL

Shiwei Yu, Xiaomeng Chen
MSc Data Science & Analytics, Georgetown University

Abstract: This project investigates the application of reinforcement learning algorithms, specifically Proximal Policy Optimization (PPO), Deep Q-Network (DQN), and Actor-Critic (A2C), in traffic simulation and navigation. The algorithms are evaluated using OpenAI's highway_env and intersection-v0 environments. Results show that DQN performs better in terms of cumulative rewards, while PPO demonstrates greater stability and performance in highway_env. However, all models exhibit weaker performance in the complex intersection-v0 environment. This highlights the challenges posed by complex action spaces. Further research is needed to develop more effective models for such environments. The findings emphasize the importance of selecting suitable algorithms for specific traffic scenarios.

Keywords: reinforcement learning, traffic simulation, navigation, PPO, DQN, A2C.

I. Introduction

The field of reinforcement learning involves tackling the challenge of traffic simulation and navigation, which entails creating simulated traffic scenarios and training intelligent agents to navigate within them. In this context, the environment is designed to mimic the behavior of vehicles, and then the agent learns to make decisions and take actions that either maximize the reward or minimize penalties. These decisions aim to optimize travel time, reduce accidents and delays, and ensure safety. This project focuses on the application of three reinforcement learning algorithms, namely Proximal Policy Optimization (PPO), Deep Q-Network (DQN), and Actor-Critic (A2C), in traffic environments specifically developed by OpenAI. These environments include a highway scenario known as "highway-env" and an intersection scenario called "intersection-env." The primary goal is to train agents capable of effectively navigating through these traffic scenarios while optimizing various reward functions. The agents must learn to take discrete actions, such as lane changes, acceleration, and braking, in order to minimize travel time, reduce accidents or delays, and prioritize safety. By evaluating and comparing the effectiveness and limitations of all three models within this context, valuable insights can be gained. This analysis will shed light on the strengths and weaknesses of each algorithm and contribute to the development of more robust and efficient techniques for traffic simulation and navigation.

II. Method

2.1 Environment

Highway_env and intersection-v0 are two traffic simulation environments developed by OpenAI, serving as the testing and evaluation platforms for reinforcement learning algorithms. These environments simulate scenarios involving multi-lane highways and intersections, respectively. The objective for the agent in both environments is to navigate without collisions while optimizing reward functions, such as minimizing travel time and reducing lane changes. The action space consists of discrete actions, including lane changes, acceleration, and braking.

2.2 RL Algorithms

Proximal Policy Optimization (PPO): Proximal Policy Optimization (PPO) is a commonly used algorithm for training agents in complex environments like autonomous driving. Its stability, robustness, and ease of tuning make it a suitable choice. In the case of Highway_env, the PPO model is trained using the stable_baselines3 library, with several hyperparameters being adjusted. The performance of the model is evaluated by generating plots of cumulative rewards versus episodes, and the mean reward is used as a metric to measure its efficiency.

Double Q-Network (DQN): Double Q-Network (DQN) is effective in handling discrete action spaces, making it a suitable algorithm for Highway_env, where the agent selects actions from a predefined set. However, compared to PPO, DQN may require more hyperparameter tuning to achieve good performance. Given that DQN can be challenging to train and may suffer from instability and slow convergence, thus the focus is on achieving faster convergence to an optimal policy by tuning hyperparameters such as buffer_size and learning_starts. Intersection-v0 is also utilized for DQN modeling.

Actor-Critic (A2C): Actor-Critic (A2C) combines the advantages of both value-based and policy-based methods. In Actor-Critic, there are two components – the actor & the critic. To apply A2C in conjunction with the intersection-v0 environment, one can define the agent using the stable_baselines3 library. This involves specifying each hyperparameter, such as the learning rate, the number of steps, the discount factor, and the number of neurons in the network. By using A2C, the agent can learn from the environment and optimize its policy to achieve better performance on the task.

III. Result and Discussion

The results of the evaluation are summarized in Table 1, which presents the cumulative, mean, and standard deviation rewards for each approach in the highway_env and intersection-v0 environments.

highway_env		intersection-v0	
PPO	DQN	DQN	Actor-Critic (A2C)
18388.01	54037.98	423.26	371.72
21.17	9.55	4.86	4.37
0.62	5.07	4.82	4.89

Table.1 Cumulative, mean, and std reward of each approach

3.1 Highway_env

The study aimed to compare the performance of Proximal Policy Optimization (PPO) and Deep Q-Network (DQN) reinforcement learning algorithms in the context of the highway_env. Performance was measured in terms of the cumulative, mean, and standard deviation of reward. The policy_kwargs=dict(net_arch=[256, 256]) parameter specified a two-layer neural network with 256 neurons in each layer for both the policy and value function in PPO and DQN. Increasing the size or complexity of the neural network can enhance the representational capacity of the policy function, potentially improving the agent's ability to learn optimal policies in the given environment. Additionally, setting train_freq = 4 helped to reduce computation during training and enhance stability by limiting the number of parameter updates, thereby preventing overfitting. The results showed that PPO outperformed DQN in terms of mean and standard deviation of reward. Specifically, PPO achieved a mean reward of 21.17 and a standard deviation of 0.62, indicating consistent and reliable performance across episodes. In contrast, DQN achieved a higher cumulative reward of 54037.98 but had higher variability in rewards earned in each episode. DQN accumulated 54038 rewards cumulatively, while PPO performed with a total reward of 18388. However, DQN's reward curve exhibited a high degree of fluctuation, resulting in a larger standard deviation of rewards compared to PPO. Despite achieving higher cumulative rewards, DQN performed worse on average due to its unstable reward curve.

Consequently, these results suggest that the choice of algorithm depends on the specific problem and its requirements. If the goal is to maximize cumulative reward, DQN may be a better choice, while if consistent performance across episodes is more important, PPO may be more suitable. Researchers and practitioners should carefully consider the trade-offs and benefits of each algorithm in the context of their particular problem.

3.2 Intersection-v0

The Deep Q-Network (DQN) and Actor-Critic (A2C) reinforcement learning models were applied in the intersection-v0 environment. Specifically, the DQN model was specified with policy_kwargs=dict(net_arch=[256, 256]), while the A2C model applied policy_kwargs=dict(net_arch=[128, 128]) with only 128 neurons in each layer. To address the value function term in the loss function, vf_coef was tuned to 0.5, which potentially led to better accuracy in estimating the value of states and ultimately better performance. However, both the DQN and A2C models performed significantly weaker in the intersection-v0 environment compared to the highway_env. The mean reward and standard deviation were relatively low, and the cumulative rewards were notably poor as well. Several factors might contribute to the poor performance in the intersection-v0 environment. Firstly, the environment itself is more complex, with a higher-dimensional state space compared to the highway_env. The agent needs to make decisions regarding the timing of starting, the appropriate speed, lane selection, stopping, and yielding to other vehicles and pedestrians. The increased complexity makes it even more challenging for the agent to learn an effective policy. Secondly, the hyperparameter settings used for these models may not be well-suited for this particular environment and may require further tuning to enhance performance. It is worth noting that hyperparameter tuning can be a time-consuming and computationally expensive process, and researchers and practitioners should carefully consider the trade-offs between the computational cost and the potential performance gains. Both DQN and A2C demonstrated a similar steady trend of increasing rewards over time,

but this trend was less steep than those observed in highway_env. This may suggest that the intersection-v0 environment is more complex, making it challenging to achieve significant differences in reward. The cumulative rewards for DQN and A2C were 423 and 371, respectively.

Based on the previous research, the DQN and A2C models performed poorly in the intersection-v0 environment, potentially due to the increased complexity of the environment and the need for further hyperparameter tuning. Future research could focus on developing more effective models or improving the hyperparameter tuning process to enhance performance in this challenging environment.

IV. Conclusion

Under the highway_env, PPO performed extremely well, demonstrating higher mean rewards and greater stability than DQN. Due to its robustness, ease of tuning, and consistent performance, PPO is a suitable choice for traffic scenarios with discrete action spaces. However, DQN achieved higher cumulative rewards but required more hyperparameter tuning and exhibited potential instability and slow convergence. In addition, under the more complex intersection-v0 environment, both the DQN and A2C models struggled to achieve good performance. The mean rewards were low in both models, and the cumulative rewards were disappointing. The complexities of the action spaces and the potential suboptimal hyperparameter tuning could explain the lower mean rewards, higher standard deviation, and poor cumulative rewards. Therefore, further exploration of hyperparameter settings and alternative algorithms is necessary to improve performance in such complex scenarios.

In conclusion, to improve the performance in complex traffic scenarios, future efforts should focus on exploring advanced algorithms and techniques. These may include hybrid approaches that combine the strengths of different reinforcement learning methods, such as incorporating elements of both PPO and DQN to leverage stability and long-term strategy learning. Furthermore, the research should also consider the utilization of more extensive hyperparameter tuning and exploration of alternative model architectures. These approaches can help to optimize the performance of the models in complex environments by effectively adapting to the specific challenges posed by higher-dimensional state spaces and various decision factors.

Reference

Farama Foundation. (2021). Intersection environment - Highway-env documentation. *Github*. Retrieved from https://farama-foundation.github.io/HighwayEnv/environments/intersection/

Kumar, P. (2019, October 20). Which Reinforcement Learning (RL) algorithm to use? Where, when, and in what scenario? *Data Driven Investor*. Retrieved from https://medium.datadriveninvestor.com/which-reinforcement-learning-rl-algorithm-to-use-where-when-and-in-what-scenario-e3e7617fb0b1

Leurent, E. (2021). Highway-env: A gym environment for testing autonomous driving agents in a highway scenario. *GitHub*. Retrieved from https://github.com/eleurent/highway-env

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2017). Continuous control with deep reinforcement learning. *Cornell University Arxiv*. Retrieved from https://arxiv.org/abs/1707.06347

Wolf, T. & Sanh, V. (2020, June 8). Introduction to Deep RL: Actor-Critic Methods. *Hugging Face*. Retrieved from https://huggingface.co/blog/deep-rl-a2c