Journal of Education and Educational Research
ISSN: 2957-9465 | Vol. 11, No. 3, 2024

Exploration of Teaching Reform in Reinforcement Learning Courses Based on Model Framework Programming

Weifeng Xu *, Mingquan Zhang, Hongtao Wang

Department of Computer, North China Electric Power University (Baoding), Baoding, China
*** Corresponding author:** Weifeng Xu

Abstract: In this paper, we summarize and analyze the problems in reinforcement learning teaching, and explores a new teaching method of reinforcement learning core algorithms based on model framework programming and through case concatenation. Firstly, we quickly construct a simulation environment for the "drone data collection" case through model framework programming. Secondly, we model the Markov decision process for this case and elaborate on the similarities and differences between deep Q-learning algorithms and actor judge algorithms. Finally, by combining case programming, we visually explain the core ideas and characteristics of the two algorithms in order to improve students' hands-on ability and enhance the teaching quality of reinforcement learning courses.

Keywords: Reinforcement Learning; Model Framework Programming; Practical Teaching.

1. Introduction

The success of deep reinforcement learning theory [1,2] in fields such as Go competition and industrial robot control has established its important position in artificial intelligence. Under the guidance of the "Action Plan for Artificial Intelligence Innovation in Higher Education Institutions", more and more universities in China are offering artificial intelligence majors and actively building an artificial intelligence curriculum system. Top universities in China, including Tsinghua University and Peking University, have successively offered reinforcement learning courses. However, research on reinforcement learning courses is still in the exploratory stage, and it is an important issue that how to design teaching models in order to enable students to master relevant theories and apply them to a specific field.

Due to the strong theoretical nature and complex mathematical derivation process of reinforcement learning courses, it is difficult for beginners to deeply understand the algorithm content and the differences between various algorithms. And, there is an urgent need to improve the integration of theory and practice. In this paper, we explore a model framework driven teaching method by combining reinforcement learning theory with practical teaching, aiming to help students better understand the mathematical theory of reinforcement learning, distinguish the similarities and differences between various algorithms, and improve their comprehensive ability to use reinforcement learning theory to solve practical problems.

2. Problems in the Teaching of Reinforcement Learning Courses

2.1. High Requirements of Reinforcement Learning Theory Teaching

Firstly, reinforcement learning is based on Markov decision processes to solve sequential decision problems, and involves knowledge of calculus, linear algebra, probability statistics, statistics, dynamic programming, Bellman optimal equations, and other related fields. The theoretical knowledge system of most students is difficult to support the learning of reinforcement learning theory. Secondly, the classification of reinforcement learning methods is complex, with overlapping knowledge points, making it difficult for students to understand and distinguish the differences between various methods. Again, reinforcement learning theory has a fast iteration and updates, which can easily lead to lagging of course knowledge points.

2.2. The Inconsistency between Theory and Practice

In the teaching of reinforcement learning theory, students often find it difficult to understand the large number of mathematical formula derivations involved in the theory, and need to deepen their understanding of reinforcement learning theory through programming practice. However, it is difficult to unify theory and practice in the teaching process. On the one hand, classic textbooks such as "Reinforcement Learning" and "Easy to Understand Reinforcement Learning" focus on the derivation and proof of formulas, but lack executable code examples, which make it difficult for students to practice. On the other hand, executable code examples are often obtained through the Internet, and the code implementation is often different from the methods introduced in the theoretical textbooks, which makes it difficult for students to understand the theory.

2.3. Lack of Practical Programming Framework in Reinforcement Learning Teaching

The deep learning frameworks mainly used in reinforcement learning teaching practice are TensorFlow and PyTorch. The syntax differences between the two frameworks are significant, and there are also many variations between different versions of the same framework, which increases the difficulty for students' practice. The practice of reinforcement learning theory requires the construction of a simulation environment, and the most commonly used simulator

currently is gym written in Python language. However, there are issues with poor scalability and disconnection from reinforcement learning algorithms during the use of gym simulator. In addition, the sample code obtained on the Internet is often aimed at a specific problem, and it is impossible to compare the differences in the calculation methods and effects of different reinforcement learning algorithms when solving the same problem, which may easily lead to deviation and confusion in students' understanding of the algorithm.

In response to the above three issues, we propose a reinforcement learning theoretical teaching method based on a model framework. combined with reinforcement learning teaching practice. The method helps students quickly build a visual simulation environment through a model framework, emphasizes the comparison of various reinforcement algorithms horizontally and deepens students' understanding of each reinforcement learning algorithm vertically, which can stimulate students' learning enthusiasm and improve the teaching quality of reinforcement learning courses from both theoretical and practical teaching perspectives.

3. Practice of Model Framework Programming in Reinforcement Learning Course Teaching

We firstly introduce the scalable model framework designed by our research group, which can help students quickly build simulation environments in different scenarios and adapt to existing mainstream reinforcement learning algorithms. Secondly, by taking the path optimization sequential decision-making problem as a case study, we constructed a simulation scenario through " UAV data collection" to reduce the teaching time and difficulty, enhance students' interest and acceptance of reinforcement learning, further exercise their hands-on ability, and deepen their understanding of reinforcement learning theory.

3.1. The Design of Model Framework

The model framework consists of two parts: simulation module and algorithm module. The simulation module can be completed in 2-4 class hours during teaching, allows students to quickly build a simulation environment for interacting with intelligent agents. The environment involves various complex models required for data collection tasks, such as agent attribute models, agent motion models, agent energy consumption models, and channel models[3]. We have implemented mainstream reinforcement learning algorithms in the algorithm module, such as deep Q-learning[4], actor critic algorithm[5], deep determination strategy gradient algorithm, etc. These algorithms can be used by teachers to quickly demonstrate algorithm training for students. We combine the case of "UAV data collection[6]" and use framework programming to reduce the difficulty of teachers' lectures, improve students' interest and hands-on ability, and further deepen students' understanding of mainstream reinforcement learning algorithm theory. The content of the UAV data collection framework is shown in Figure 1.

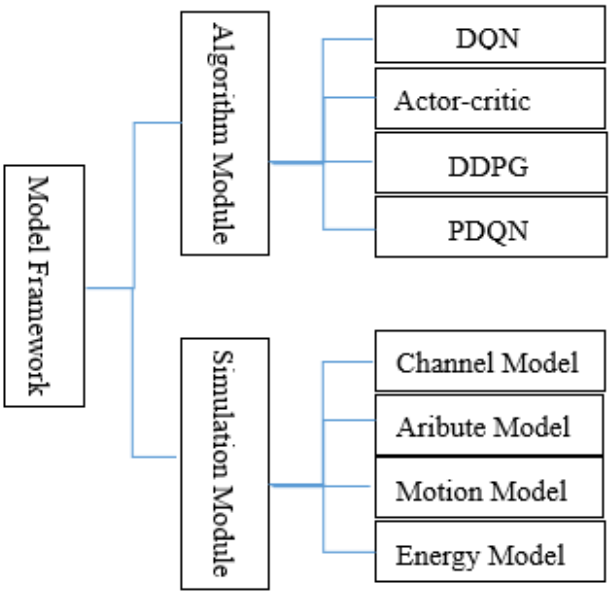

Figure 1. UAV data collection framework

3.2. The Case Design

We select a 50 × 50 area for the case of "UAV data collection" as the data collection range of the UAV [7], which stores 10 fixed nodes to be collected. The UAV takes off from the initial position and sequentially accesses the nodes collected to obtain data. The goal of the "UAV data collection" task is to find the optimal flight route so that the UAV can complete the data collection task within the specified time with minimal energy consumption.

We construct a Markov decision process model for the above case. The state space includes two parts: the status of the nodes collected and the status of the UAV. The former includes information such as the location and data volume of the nodes collected, while the latter includes information about the status of the UAV, such as location and energy. In this case, the action space consists of 10 discrete actions, representing each node to be collected. The instantaneous reward is designed based on factors that affect the algorithm's objectives, such as the drone's flight distance and time.

We can quickly implement simulation scenarios and construct Markov decision models based on the drone framework. On this basis, we will teach reinforcement learning methods based on value and represented by deep Q-learning and reinforcement learning methods based on strategy and represented by actor judge algorithm, focus on the core theories and differences between the two algorithms. Then, we will guide students to design reward functions and program them based on the corresponding algorithm theories, ultimately achieve optimal UAV flight routes [8,9].

3.3. The Algorithm Theory

Reinforcement learning mainly solves sequential decision-making problems, where agents observe the environment and select and execute corresponding actions based on algorithms, and then enter the next state while obtaining an instantaneous reward. We propose to design an example of "unmanned aerial vehicle data collection" based on the programming concept of model framework, and discuss the deep Q-learning algorithm and actor critic algorithm. The two algorithm theories and their differences are introduced as follows.

3.3.1. Deep Q-learning Algorithm

Deep Q-learning algorithm is a typical reinforcement

learning algorithm based on value [1]. This algorithm is based on the Bellman equation, evaluates the action value function corresponding to each action of the agent in the current state S, and selects the action to be executed by the agent according to the probability distribution. Deep Q-learning uses the idea of temporal difference method to train neural networks, breaks action correlation and alleviate algorithm bootstrap problem with experience pool and target network to, which can accelerate the convergence speed of the algorithm.

3.3.2. Actor Judge Algorithm

The actor critic algorithm is a typical reinforcement learning algorithm based on strategy [2,5]. This algorithm maximizes the expected value of the state value function based on the policy gradient theorem. The algorithm constructs an actor network and a judge network. The former is used for the agent to select actions based on the current state, while the latter scores and evaluates the actions made by the agent.

3.4. Associated Programming

In the case of "UAV data collection", students first apply the model framework to build a simulation environment. This process includes defining the node model to be collected (including attribute information such as quantity, location, and initial data volume), loading the energy consumption model and data generation model, defining the drone model (including attribute information such as location and energy), loading the collection model, motion model, and energy consumption model [10]. Then, students construct a Markov model for the problem and implement the code of neural network construction, reward function, and loss function based on the theories of deep Q-learning algorithm and actor judge algorithm. Due to the use of two types of algorithms for programming the same case, there is no need to modify the environment part of the code, which allows for a more intuitive display of the similarities and differences between the two algorithms.

Although the deep Q-learning algorithm and the actor judge algorithm have similar network structures in the decision-making part, their actual meanings expressed are completely different. The output of the deep Q-learning algorithm neural network predicts the action value function corresponding to each action, and the agent evaluates the quality of the action based on this value and selects the better action to execute according to probability. The output of the actor judge algorithm neural network is only the probability of selecting each action, and the quality of the action is evaluated through the judge neural network. The actor modifies the action selection probability in each state according to the requirements of the judge. Both algorithms have bootstrap problems and use the target network method to optimize the model training process.

In previous teaching, students tend to confuse the theories of two reinforcement learning algorithms. In this paper, we introduce the case of "UAV data collection". By quickly constructing a reusable reinforcement learning simulation environment based on the model framework, we can compare the similarities and differences between two algorithms in the same environment, intuitively teach the core ideas of the two algorithms, which reduce the difficulty of teaching for teachers, deepen students' understanding of algorithms, and solves the problem of the inconsistency between algorithm theory and practice. In addition, students can use the model framework proposed in this paper for self-directed learning. They can expand the model and reinforcement learning algorithms according to their needs, and enhance their confidence and programming ability.

4. Conclusion

In this paper, we elaborate on the teaching method of introducing model framework programming into reinforcement learning teaching, and use the case of "UAV data collection" to illustrate the similarities and differences between deep Q-learning algorithm and actor judge algorithm. The case is simple and clear, and students can use the model framework to quickly build a reinforcement learning environment. By applying two algorithms in the same environment, students can intuitively learn and compare the core ideas of the two algorithms, which effectively improve their programming skills and enhance the teaching quality of reinforcement learning courses.

References

[1] Allah Mottaki N, Motameni H, Mohamadi H. A genetic algorithm-based approach for solving the target Q-coverage problem in over and under provisioned directional sensor networks[J/OL]. Physical Communication, (2021-01-01)[2023-01-08]. https:// doi.org/10. 1016/j.phycom. 2022. 101719.

[2] Yang Xiao. Research and application of key technologies for autonomous network based on deep reinforcement learning [D]. Beijing University of Posts and telecommunications, 2024.DOI:10. 26969/d.cnki.gbydu.2024. 000285.

[3] Zhang Guangchi, He Zinan, Cui Miao. Energy Consumption Optimization of Unmanned Aerial Vehicle Assisted Mobile Edge Computing Systems Based on Deep Reinforcement Learning[J]. Journal of Electronics & Information Technology, 2023,45(05):1635-1643. doi: 10.11999/JEIT220352.

[4] Cao X, Xu W, Liu X, et al. A deep reinforcement learning-based on-demand charging algorithm for wireless rechargeable sensor networks[J/OL]. Ad Hoc Networks,, (2021-01-01)[2023-01-08]. https://doi.org/10.1016/j.adhoc.2020.102278.

[5] Liheng Lv. High-Accuracy Non-Intrusive Load Monitoring Algorithms Based on Deep Learning [D]. Jilin University, 2024.

[6] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning[J/OL]. arXiv preprint, (2013-12-19) [2023-01-08]. https://doi.org/10.48550/arXiv.1312.5602.

[7] Liu K, Zheng J. UAV trajectory optimization for time-constrained data collection in UAV-enabled environmental monitoring systems[J]. IEEE Internet of Things Journal, 2022, 9(23): 24300-24314. doi: 10.1109/JIOT.2022.3189214.

[8] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” IEEE Commun. Mag., vol. 54, no. 5, pp. 36–42, May 2016.

[9] Y. Zeng, J. Xu, and R. Zhang, “Energy minimization for wireless communication with rotary-wing UAV,” IEEE Trans. Wireless Commun., vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[10] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, “IoT enabled UAV: Network architecture and routing algorithm,” IEEE Internet Things J., vol. 6, no. 2, pp. 3727–3742, Apr. 2019.