

An Introduction to Reinforcement Learning: Key Concepts, Algorithms and Applications

Alexander Yurchenkov*

Department of Control Sciences of Russian Academy of Sciences, University of Bauman Moscow State Technical, Moscow, Russia

Introduction

Reinforcement Learning (RL) is a subfield of Machine Learning (ML) that focuses on developing algorithms and models that enable an agent to learn from its environment through trial and error, by maximizing a numerical reward signal. In other words, RL is an approach to AI that allows an agent to learn how to behave in an environment by interacting with it, in order to achieve a specific goal. This goal can be anything from winning a game to controlling a robot arm. The basic idea behind RL is to create an environment in which an agent can learn by taking actions and receiving feedback in the form of rewards or punishments. The agent's objective is to maximize the cumulative reward it receives over time. The agent does this by learning a policy, which is a mapping from states to actions that maximizes the expected cumulative reward. The policy can be either deterministic or stochastic [1].

Description

Let's take a closer look at each of these components. The agent is the entity that interacts with the environment. It takes actions based on the state of the environment and receives feedback in the form of rewards or punishments. The goal of the agent is to maximize the cumulative reward it receives over time. The environment is the external system that the agent interacts with. It can be anything from a simulated world to a physical system. The environment provides feedback to the agent in the form of rewards or punishments. The state is the current configuration of the environment. It is a snapshot of the environment at a given time. The state can be either observable or partially observable. In the case of observable state, the agent has full access to the state of the environment. In the case of partially observable state, the agent has limited access to the state of the environment. The action is the decision made by the agent based on the current state of the environment. The action can be discrete or continuous. In the case of discrete action, the agent can choose from a finite set of actions [2].

In the case of continuous action, the agent can choose from an infinite set of actions. The reward is the feedback provided by the environment to the agent. It can be positive, negative, or neutral. The goal of the agent is to maximize the cumulative reward it receives over time. The policy is the mapping from states to actions that maximizes the expected cumulative reward. The policy can be either deterministic or stochastic. In the case of deterministic policy, the agent always chooses the same action given a particular state. In the case of stochastic policy, the agent chooses actions randomly based on a probability distribution. Now that we have a basic understanding of the components of RL, let's take a look at some of the key algorithms and techniques used in RL [3].

Q-Learning is a popular RL algorithm that uses a value function to estimate

the expected cumulative reward of taking an action in a given state. The value function is used to update the policy by selecting the action with the highest expected cumulative reward. Policy Gradient is a technique used to learn a stochastic policy by directly optimizing the expected cumulative reward. The policy is updated by computing the gradient of the expected cumulative reward with respect to the policy parameters [4,5].

Conclusion

Actor-Critic is a hybrid RL algorithm that combines elements of both Q-Learning and Policy Gradient. It uses a value function to estimate the expected cumulative reward and a policy function to select actions. Deep Reinforcement Learning (DRL) is a variant of RL that uses deep neural networks to represent the value function and policy function. DRL has been successfully applied to a wide range.

Acknowledgement

None.

Conflict of Interest

There are no conflicts of interest by author.

References

1. Kang, Daeshik, Peter V. Pikhitsa, Yong Whan Choi and Chanseok Lee, et al. "Ultrasensitive mechanical crack-based sensor inspired by the spider sensory system." *Nature* 516 (2014): 222-226.
2. Ali, Farhan, Timothy M. Otchy, Cengiz Pehlevan and Antoniu L. Fantana, et al. "The basal ganglia is necessary for learning spectral, but not temporal, features of birdsong." *Neuron* 80 (2013): 494-506.
3. Van der Laan, Mark and Alexander Luedtke. "Targeted learning of the mean outcome under an optimal dynamic treatment rule." *J Causal Infer* 3 (2015): 61-95.
4. Russ, John C., James R. Matey, A. John Mallinckrodt and Susan McKay, et al. "The image processing handbook." *Phys Comput* 8 (1994): 177-178.
5. Chhetri, Tek Raj, Anelia Kurteva, Jubril Gbolahan Adigun and Anna Fensel, et al. "Knowledge graph based hard drive failure prediction." *Sensors* 22 (2022): 95-100.

*Address for Correspondence: Alexander Yurchenkov, Department of Control Sciences of Russian Academy of Sciences, University of Bauman Moscow State Technical, Moscow, Russia; E-mail: Yu.alexander@gmail.com

Copyright: © 2023 Yurchenkov A. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Received: 30 April, 2023, Manuscript No. sndc-23-96037; Editor Assigned: 02 May, 2023, PreQC No. P-96037; Reviewed: 15 May, 2023, QC No. Q-96037; Revised: 22 May, 2023, Manuscript No. R-96037; Published: 30 May, 2023, DOI: 10.37421/2090-4886.2023.12.209

How to cite this article: Yurchenkov, Alexander. "An Introduction to Reinforcement Learning: Key Concepts, Algorithms and Applications." *Int J Sens Netw Data Commun* 12 (2023): 209.