**1. Basic Information**

* **Title: Introduction to Reinforcement Learning**
* **Authors: Majid Ghasemi⋆,1 and Dariush Ebrahimi1**
* **Published in:** *arXiv preprint arXiv* 2024
* **Link:** <https://arxiv.org/pdf/2408.07712>
* **Group:** Fall 2025
* **Reviewer:** Ju Esther

**2. Motivation & Problem**

* 연구 동기: To simplify the complexities of RL for beginners, providing a straightforward pathway to understanding and applying real-time techniques

**3. Result**

* 기본용어

State: a specific condition or configuration of the environment at a given time as perceived by the agent

Action: set of possible moves or decisions an agent can make while interacting with the environment

Policy: guides the behavior of a learning agent by mapping perceived states of the environment into actions

Rewards: provides the agent with an objective at each time step, defining both local and global goals that the agent aims to achieve over time.

Transition dynamics: the probability of reaching a new state s′ given the current state s and action a

* Key concepts

Multi-Armed Bandit(s): a simplified decision-making scenario where an agent repeatedly chooses from K actions (or arms) to maximize cumulative rewards over time. Each action is associated with an unknown reward distribution, requiring the agent to balance exploration (gathering information about all actions) and exploitation (maximizing immediate rewards using the best-known action).

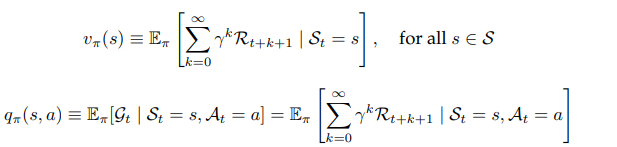


MarkovDecision Process (MDPs): actions affect immediate rewards as well as future outcomes. Aim to measure the value of taking action a in state s, or the value of being in state s assuming optimal actions are taken. A



Policies and Value Functions: estimates the expected return of the agent being in a certain state.

* State Value Functions: In the evaluation of deterministic policies or when understanding the value of being in a particular state is required
* Action Value Functions: evaluate and compare the potential for different actions when they are taking place in the same state.



Optimal Policies and Optimal Value Functions: maximizes long-term rewards.



텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Policy Evaluation (Prediction): assesses the expected return when following policy π from each state.



Policy Improvement: identify improved policies. Compare the value of taking a different action a in state s with the current policy



Policy Improvement Theorem: creates a new policy that enhances an initial policy by adopting a greedy approach based on the value function.

Policy Iteration: a sequence of improving policies and corresponding value functions. Converges to an optimal policy and value function 9 in a finite number of iterations



Value Iteration:

Policy iteration limitation: each iteration requires policy evaluation, often necessitating multiple passes through the entire state set

* policy evaluation can be abbreviated without losing convergence guarantees
* terminates policy evaluation after a single sweep

efficiency, can be implemented using a synchronous update approach, its robustness to initial conditions(reliable), provides a foundation for more advanced algorithms

* Core RL mothods

Model-free & Model-based methods

* Model-free: determine the optimal policy or value function directly without constructing a model of the environment. (Value based, policy based, hybrid)
* Model-based: predict the outcomes of actions, which facilitate strategic planning and decision-making. Enhance learning efficiency by providing opportunities for virtual experimentation, despite the complexity

Off-Policy and On-Policy Methods

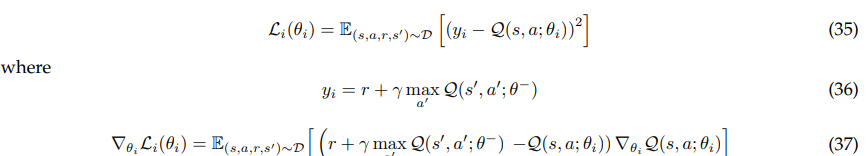
* On-policy: evaluate and improve the policy used to make decisions, intertwining exploration and learning.
* Off-policy: involve learning the value of the optimal policy independently of the agent’s actions.
* Behavior policy: strategy used by an agent to determine which actions to take at each time step
* target policy: how the agent updates its value estimates in response to observed outcomes
* Essential algorithms

Value-based:

* Q-learning: a Model-free algorithm considered as off-policy Temporal Difference (TD) control.



* Deep Q-Networks (DQN): merges Q-learning with Neural Networks to learn control policies directly from raw pixel inputs



Policy-based: more strongly emphasizes direct policy optimization in the process of choosing actions for an agent. Better dealing with very challenging environments that have high-dimensional action spaces or where policies are inherently stochastic.

* Reinforce: a seminal contribution to RL, particularly within the context of policy gradient methods.

스크린샷, 텍스트, 폰트, 블랙이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

* Proximal Policy Optimization: Aims to achieve reliable performance and sample efficiency, addressing the limitations of previous policy optimization algorithms



Hybrid (Actor-Critic) methods: combine Value-based and Policy-based approaches.

* Asynchronous Advantage Actor-Critic (A3C) and Advantage Actor-Critic (A2C)



**4. References & Resources**

* 원 논문 <https://arxiv.org/pdf/2408.07712>

**1. Basic Information**

* **Title: Understanding Reinforcement Learning Algorithms: The Progress from Basic Q-learning to Proximal Policy Optimization**
* **Authors: Mohamed-Amine Chadia\*, Hajar Mousannif**
* **Published in:** *arXiv preprint* (2023)
* **Link:** [2304.00026](https://arxiv.org/pdf/2304.00026)
* **Group:** Fall 2025
* **Reviewer:** Esther Ju

**2. Motivation & Problem**

* 연구 동기: RL can be a challenging field for beginners to understand, aims to serve as a valuable resource for beginners

**3. Key Result**

* Common Background
* Exploration-exploitation trade-off
* Explore the environment and try different actions to learn which actions lead to higher rewards.
* Exploit its current knowledge by taking actions that are expected to lead to the highest reward based on its current policy.
* Key terms

Agent: a learning-based model

Environment: simulation

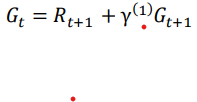
Action

Reward

State

Policy: sequence of actions executed by the agent given the corresponding environment’s states

* Markov decision process: {S, A, T(Transition function), R, γ(Discount factor)}.
* Bellman equation trick: proposed a recursive approximation of the return G and the value V



* Algorithms: motivation, inner workings, limitation
* Q-learning (value-based): considers a variant of this, called Q-function, where Q designates quality.



Update q-value



Limitation: applied in limited applications, since it uses finite-size tables

* Deep Q-learning (value-based):

기존 Q-learning 문제: The experience correlation, The moving target problem

* 해결: The replay memory, The target network

Limitation: categorical way by which actions are generated, makes it unable to be used in continuous action settings

* Reinforcement(policy-gradient): The gradient loss in PG algorithms is directly proportional to the policy in an intuitive way



Limitation: it is not clear which action resulted in which reward.

* DDPG (actor-critic): an actor-critic model

Actor(learning policy)

Critic(learning the value function of each state (V) or state-action pair (Q))

알고리즘 4 “Store transition (st , at , rt ,st+1 ) in D” replay memory 사용

Limitation: unstable.

* TD3 (actor-critic): solves the overestimation bias in the DDPG model issue by maintaining a pair of critics Q1 and along with a single actor, and each network with its corresponding target.

Uses the smaller of the two Q-values and updates the policy less frequently than the critic networks.

Limitation: handled manually with a gaussian noise, computationally heavy, only be used naturally with continuous space environments.

* PPO (actor-critic): collects mini batches of experience while interacting with the environment and use them to update its policy.

문제: trust region

해결: ease of implementation • sample efficiency • ease of tuning

Limitation: not explored

**4. References & Resources**

* 원 논문 <https://arxiv.org/pdf/2304.00026>