Навчання з підкріпленням

Лекція 8: Методи градієнту стратегії

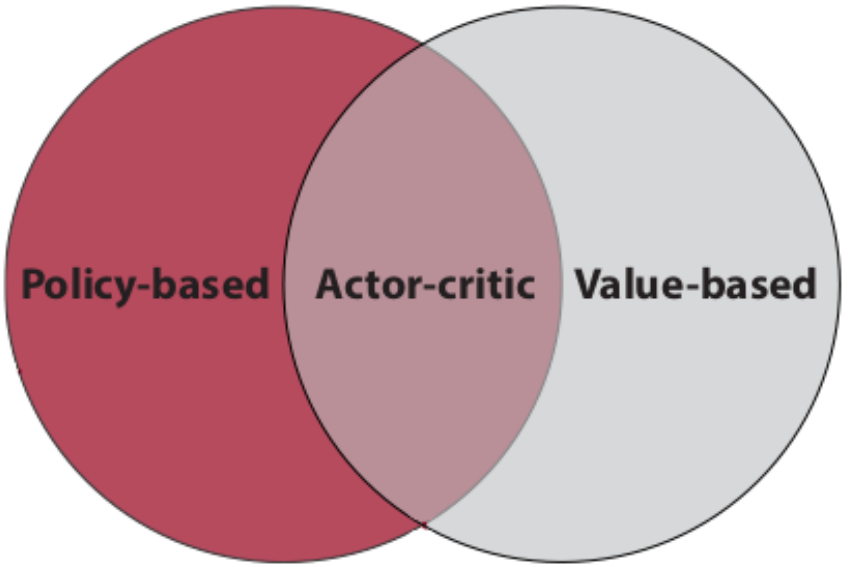
Кочура Юрій Петрович
iuriy.kochura@gmail.com
@y_kochura

Сьогодні

- Градієнтні методи стратегії

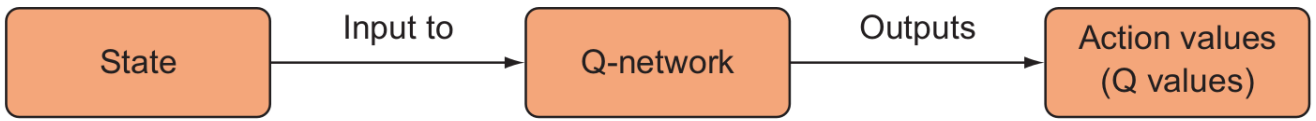
Вступ

Policy-based, value-based, and actor-critic methods



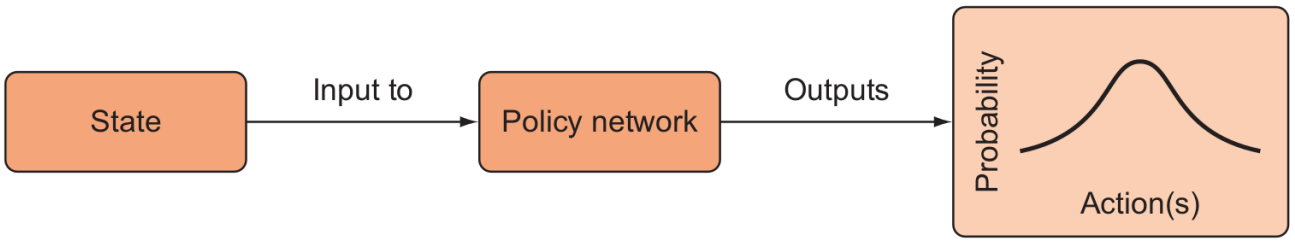
—

Джерело: Grokking Deep Reinforcement Learning, Miguel Morales.



—

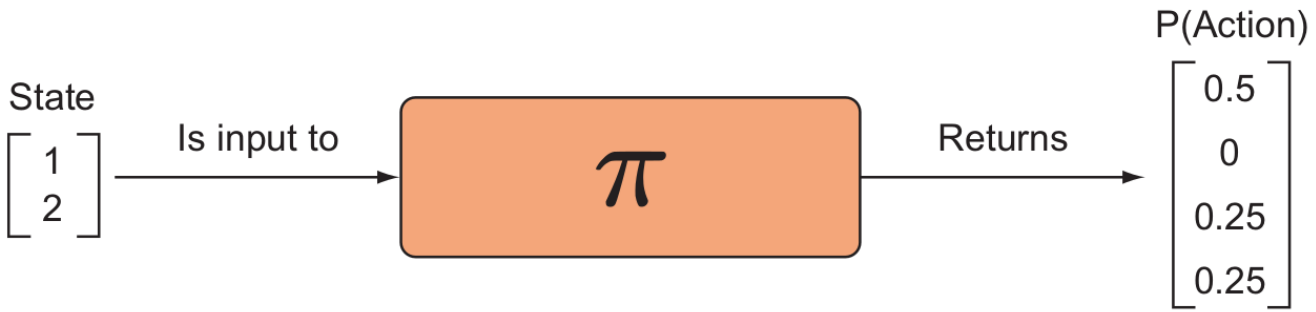
Джерело: Deep Reinforcement Learning in Action, Alexander Zai and Brandon Brown.



—

Джерело: Deep Reinforcement Learning in Action, Alexander Zai and Brandon Brown.

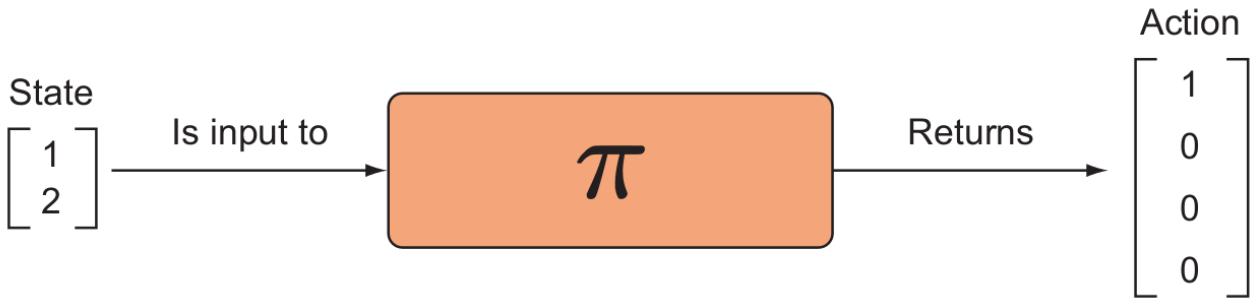
Стохастична стратегія



—

Джерело: Deep Reinforcement Learning in Action, Alexander Zai and Brandon Brown.

Детермінована стратегія



—

Джерело: Deep Reinforcement Learning in Action, Alexander Zai and Brandon Brown.

Цільова функція

Show Me the Math

Value-based vs. policy-based methods objectives

(1) In value-based methods, the objective is to minimize the loss function, which is the mean squared error between the true Q-function and the parameterized Q-function.

$$L_i(\theta_i) = \mathbb{E}_{s,a}\left[\left(q_\pi(s,a) - Q(s,a;\theta_i)\right)^2\right]$$

$$J_i(\theta_i) = \mathbb{E}_{s_0 \sim p_0}\left[v_{\pi_{\theta_i}}(s_0)\right]$$

(2) In policy-based methods, the objective is to maximize a performance measure, which is the true value-function of the parameterized policy from all initial states.

—

Джерело: Grokking Deep Reinforcement Learning, Miguel Morales.

Цільова функція

- Goal: given policy $\pi_\theta(s, a)$ with parameters θ, find best θ
- But how do we measure the quality of a policy π_θ?
- In episodic environments we can use the start value

$$J_1(\theta) = V_{\pi_\theta}(s_0) = \mathbb{E}[v_{\pi_\theta}(s_0)]$$

- In continuing environments we can use the average value

$$J_{av}(\theta) = \sum_s d_{\pi_\theta}(s) V_{\pi_\theta}(s)$$

- Or the average reward per time-step

$$J_{avR}(\theta) = \sum_s d_{\pi_\theta}(s) \sum_a \pi_\theta(s, a) \mathcal{R}_s^a$$

- where $d_{\pi_\theta}(s)$ is stationary distribution of Markov chain for π_θ

—

Джерело: David Silver.

Value-based vs. policy-based vs. policy-gradient vs. actor-critic methods

- **Value-based methods:** Refers to algorithms that learn value functions and only value functions. Q-learning, SARSA, DQN are all value-based methods.
- **Policy-based methods:** Refers to a broad range of algorithms that optimize policies, including black-box optimization methods, such as genetic algorithms.
- **Policy-gradient methods:** Refers to methods that solve an optimization problem on the gradient of the performance of a parameterized policy.
- **Actor-critic methods:** Refers to methods that learn both a policy and a value function, primarily if the value function is learned with bootstrapping and used as the score for the stochastic policy gradient.

—

Джерело: Grokking Deep Reinforcement Learning, Miguel Morales.

Policy Optimisation

- Policy based reinforcement learning is an optimisation problem
- Find θ that maximises $J(\theta)$
- Some approaches do not use gradient
  - Hill climbing
  - Simplex / amoeba / Nelder Mead
  - Genetic algorithms
- Greater efficiency often possible using gradient
  - Gradient descent
  - Conjugate gradient
  - Quasi-newton

—
Джерело: David Silver.

Advantages of Policy-Based RL

Advantages:

- Better convergence properties
- Effective in high-dimensional or continuous action spaces
- Can learn stochastic policies

Disadvantages:

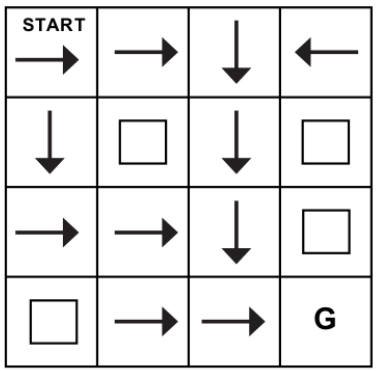
- Typically converge to a local rather than global optimum
- Evaluating a policy is typically inefficient and high variance

—

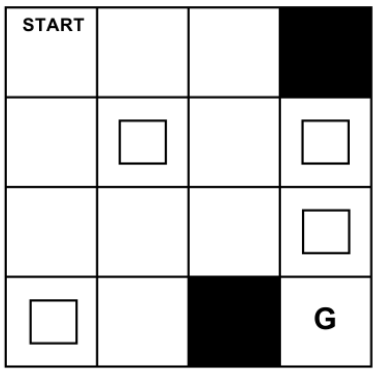
Джерело: David Silver.

Learning stochastic policies could get us out of trouble

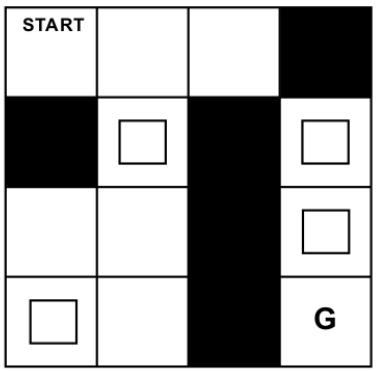
(1) Consider a foggy lake environment in which we don't slip as in the frozen lake, but instead we can't see which state we're in.



(2) If we could see well in every state, the optimal policy would be something like this.



(3) If we couldn't see in these two states, the optimal action in these states would be something like 50% left and 50% right.



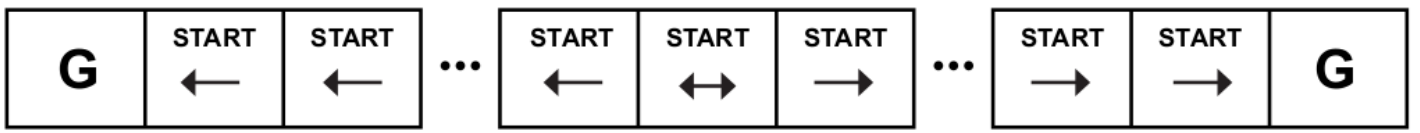
(4) The more partially observable, the more complex the probability distribution to learn for optimal action selection.

—

Джерело: Grokking Deep Reinforcement Learning, Miguel Morales.

Learning policies could be an easier, more generalizable problem to solve

(1) Consider a near-infinite corridor deterministic environment in which there's a large number of cells, say 1,000,001. There are two goals, one in the leftmost cell, the other in the rightmost cell, and every non-terminal state is in the set of initial states.



(2) In an environment like this, the optimal policy would look as shown here. In the middle cell, cell 500,000, a 50% left and a 50% right is optimal. The rest of the actions should point to the closest goal.

(3) The optimal policy in this environment is rather obvious, but what isn't so obvious is that learning and generalizing over policies is likely easier and more straightforward than learning value functions. For instance, do I care whether cell 1000 is 0.0001 or 0.00014 or anything else, if the action is obviously left? Allocating resources for accurately estimating value functions is unlikely to yield any advantages over discovering the pattern over actions.

—

Джерело: Grokking Deep Reinforcement Learning, Miguel Morales.

Show Me the Math

Deriving the policy gradient

(1) First, let's bring back a simplified version of the objective equation a couple of pages ago.

$$J(\theta) = \mathbb{E}_{s_0 \sim p_0}\left[v_{\pi_\theta}(s_0)\right]$$

(2) We know what we want is to find the gradient with respect to that performance metric.

$$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{s_0 \sim p_0}\left[v_{\pi_\theta}(s_0)\right]$$

(3) To simplify notation, let's use tau as a variable representing the full trajectory.

$$\tau = S_0, A_0, R_1, S_1, ..., S_{T-1}, A_{T-1}, R_T, S_T$$

(4) This way we can abuse notation and use the G function to obtain the return of the full trajectory.

$$G(\tau) = R_1 + \gamma R_2 +, ..., +\gamma^{T-1} R_T$$

(5) We can also get the probability of a trajectory.

(6) This is the probability of the initial states, then the action, then the transition, and so on until we have the product of all the probabilities that make the trajectory likely.

$$p(\tau|\pi_\theta) = p_0(S_0)\pi(A_0|S_0;\theta)P(S_1, R_1|S_0, A_0)...P(S_T, R_T|S_{T-1}, A_{T-1})$$

(7) After all that notation change, we can say that the objective is this.

$$\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta}\left[G(\tau)\right] = \nabla_\theta \mathbb{E}_{s_0 \sim p_0}\left[v_{\pi_\theta}(s_0)\right]$$

(8) Next, let's look at a way for estimating gradients of expectations, called the score function gradient estimator.

$$\nabla_\theta \mathbb{E}_x\left[f(x)\right] = \mathbb{E}_x\left[\nabla_\theta \log p(x|\theta) f(x)\right]$$

(9) With that identity, we can substitute values and get this.

$$\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta}\left[G(\tau)\right] = \mathbb{E}_{\tau \sim \pi_\theta}\left[\nabla_\theta \log p(\tau|\pi_\theta) G(\tau)\right]$$

(10) Notice the dependence on the probability of the trajectory.

(11) If we substitute the probability of trajectory, take the logarithm, turn products into the sum, and differentiate with respect to theta, all dependence on the transition function is dropped, and we're left with a function that we can work with.

$$\nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta}\left[G(\tau)\right] = \mathbb{E}_{\tau \sim \pi_\theta}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi(A_t|S_t;\pi_\theta) G(\tau)\right]$$

—

Джерело: Grokking Deep Reinforcement Learning, Miguel Morales.

Література

- David Silver, Lecture 7: Policy Gradient Methods. [video], [slides]
- Reinforcement Learning: An Introduction - Chapter 13: Policy Gradient Methods
- Reinforcement Learning Series: Overview of Methods

Кінець