FPT Education
FPT UNIVERSITY

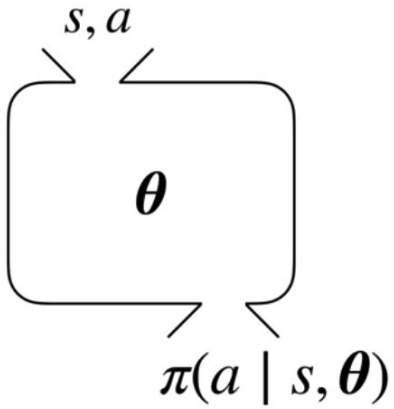
Learning Parameterized Policies

Objectives

- Understand Parameterized Policies
- Understand Softmax policy parameterization
- Understand the Advantages of Policy Parameterization

Parameterizing Policies Directly

- Using the Greek letter Theta for the policies parameter vector.
- Parameterized policy: given the input state and action, the parameterized policy function will output the probability of taking that action in that state.



Parameterizing Policies Directly

- Constraints on the Policy Parameterization
  - The parameterized function has to generate a valid policy.
  - The probabilities selecting an action, must be greater than or equal to zero.
  - For each state, the sum of the probabilities over all actions must be one.

$$\pi(a \mid s, \boldsymbol{\theta}) \geq 0 \qquad \text{for all } a \in \mathcal{A} \text{ and } s \in \mathcal{S}$$

$$\sum_{a \in \mathcal{A}} \pi(a \mid s, \boldsymbol{\theta}) = 1 \qquad \text{for all } s \in \mathcal{S}$$

Softmax Policy Parameterization

- Example of softmax policies
  - The function h is called the action preference.
  - Computing the probability of selecting an action with the softmax: we take the action preference, exponentiate it, and then divide by the sum over all the actions for the same thing

$$\pi(a \mid s, \boldsymbol{\theta}) \doteq \frac{e^{h(s,a,\boldsymbol{\theta})}}{\sum_{b \in \mathcal{A}} e^{h(s,b,\boldsymbol{\theta})}}$$

Softmax Policy Parameterization

- Action preferences are not action values

Advantages of Policy Parameterizatio

- Compact Representation:
  - Parameterized policies allow for compact representations of complex policies.
- Generalization:
  - Parameterized policies facilitate generalization across similar states.
- Continuous Action Spaces:
  - Parameterized policies can easily handle continuous action spaces.

Advantages of Policy Parameterizatio

- Flexibility:
  - Parameterized policies offer flexibility in representing a wide range of policies.
- Learning Complex Policies:
  - Parameterized policies are well-suited for learning complex policies in high-dimensional state spaces.
- End-to-End Learning:
  - Parameterized policies support end-to-end learning, where the policy function is directly optimized to maximize cumulative rewards.

Advantages of Policy Parameterizatio

- Policy parameterization offers a powerful framework:
  - Representing and Learning policies in reinforcement learning,
  - Enabling agents to effectively navigate complex environments
  - Achieve high-performance outcomes.

The Softmax Policy example

- Using a softmax policy to select actions in a simple grid world environment.

```python
# 3.9 The softmax policy example-HoaDNT@fe.edu.vn
import numpy as np

class SoftmaxPolicy:
    def __init__(self, num_actions, temperature):
        self.num_actions = num_actions
        self.temperature = temperature

    def select_action(self, q_values):
        probabilities = self.softmax(q_values)
        action = np.random.choice(self.num_actions, p=probabilities)
        return action

    def softmax(self, q_values):
        scaled_values = q_values / self.temperature
        exp_values = np.exp(scaled_values)
        probabilities = exp_values / np.sum(exp_values)
        return probabilities
```

The Softmax Policy example

```
19
20 # Example usage:
21 num_actions = 4  # Up, Down, Left, Right
22 temperature = 0.5  # Softmax temperature
23
24 # Initialize a softmax policy
25 softmax_policy = SoftmaxPolicy(num_actions, temperature)
26
27 # Example Q-values (arbitrary values for illustration)
28 q_values = np.array([1.0, 2.0, 0.5, 1.5])
29
30 # Select an action based on the softmax policy
31 selected_action = softmax_policy.select_action(q_values)
32 print("Selected Action:", selected_action)
33
```

```
Selected Action: 1
```

Summary

- Understand Parameterized Policies
- Understand Softmax policy parameterization
- Understand the Advantages of Policy Parameterization

Q & A