Solving the Cartpole Control Problem with Tabular Q-Learning

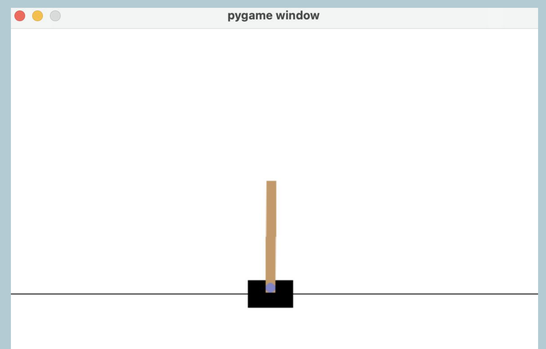
New College of Florida
The Honors College

Bilge Akyol
Reinforcement Learning Fall 2025

OpenAI Gymnasium

Introduction

CartPole is a small cart that moves in a straight line. A thin pole is attached to the cart. The pole is affected by gravity and falls. Our role is to teach an RL agent to move the cart so that the pole stays straight up.



Background & Motivation

CartPole is frequently used to test learning algorithms because it is simple, interpretable, and requires real-time decision-making based on continuous states. For the pole to stay vertical, the cart must move left or right. The pole that is swinging on the cart has **continuous** observations (cart position, speed, pole angle, angular velocity).

Q-Learning is an effective off-policy TD method that directly updates the Q-values of state-action pairs at each step, eventually converging to the optimal Q-value. However, it is naturally suited for **discrete** state spaces, so applying it directly to continuous environments requires **discretization**.

Motivations:

- To work with Gymnasium environment
- To work with discretizing continuous states to match algorithms that use discrete states
- To implement a tabular Q-Learning agent
- To compare different parameters and optimizations to compare efficiency and accuracy.
- To practice Q-Learning update rule on a practical project

Methodology

The agent doesn't need to understand the full environment. It needs to be consistent. With each step, its decisions get a little better. Every time the agent takes an action and receives feedback. After this step, it updates its internal memory – the Q-table. It's like a person who adjusts their habits after each small success or failure.

Since the goal is to keep the pole upright for as long as possible, by default, a reward of +1 is given for every step taken, including the termination step. The default reward threshold is 500 for v1 (the one we use in this project) and 200 for v0 due to the time limit on the environment.

1. Environment
   - OpenAI Gymnasium CartPole-v1.
2. State Discretization
   - Each dimension divided into 12 bins using np.linspace.
   - Velocity and angular velocity bounds manually clipped to [−3, 3] to avoid infinite ranges.
3. Q-Table
   - 5-dimensional array: [12,12,12,12,2]
   - initialized to zeros
4. Training
   - Q-learning with epsilon-greedy exploration, with epsilon decaying over time
   - Episodes: 25,000.
   - Hyperparameters:
     - $\alpha = 0.1$
     - $\gamma = 0.99$
     - ε decreasing from 1.0 → 0.01
     - Reward: +1 per timestep alive
5. Testing
   - After training, $\varepsilon = 0$.
   - Run 5 episodes with deterministic (greedy) policy
   - Render

Q-Learning Update Rule:

$$Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$$

Environment

CartPole-v1 environment in Gymnasium

This environment corresponds to the version of the cart-pole problem described by Barto, Sutton, and Anderson.

Each observation is a real-valued vector: $[x, \dot{x}, \theta, \dot{\theta}]$
Cart Position (x): Horizontal position of the cart on the track.

Cart Velocity ($\dot{x}$): Rate of movement of the cart left or right.

Pole Angle (θ): Angle of the pole relative to vertical (0 = perfectly upright).

Pole Angular Velocity ($\dot{\theta}$): Rate at which the pole angle is changing.

Hyperparameters

alpha = 0.1 — — — — — — Learning Rate
gamma = 0.99 — — — — — — Discount Factor
epsilon = 1.0 — — — — — — Exploration
ε min = 0.01
epsilon_decoy
Reward: +1 per timestep alive

Results

- The agent successfully learned to balance the pole consistently.
- Convergence to near-optimal performance (> 450 average reward)
- Final test runs typically achieve reward close to the maximum of 500, indicating successful control.
- Discretization with 12 bins proved sufficient for stable learning without requiring function approximation.
- Achieved stable pole control!

```
Test episode reward: 500.0
Test episode reward: 500.0
Test episode reward: 500.0
Test episode reward: 500.0
Test episode reward: 500.0
```

Conclusion

- Tabular Q-learning combined with careful discretization can solve the continuous CartPole-v1 environment to high performance.
- Using uniform bins, clipped velocity bounds, and a moderately large number of bins (25 instead of 10) improves state representation enough for convergence
- Although not as versatile as deep RL methods like DQN, Q-learning produces stable policies for **low-dimensional** control tasks and simple tabular methods remain effective.

Discussion

Future Work

- Implement DQN to handle continuous states without discretization
- Try tile coding or finer discretization
- Perform hyperparameter search