Reinforcement Learning Implementation

Carlos Rodriguez

December 2024

1 Introduction

This project implements core reinforcement learning principles to solve complex sequential decision-making problems across three distinct environments. Using value function approximation techniques, we demonstrate how an agent can learn optimal policies through direct interaction with its environment, without any prior knowledge of the environment dynamics.

The implementation focuses on fundamental RL concepts: state-action value estimation, policy improvement through exploration and exploitation, and reward-driven learning. Through careful environment design and reward shaping, we show how agents can learn effective policies in environments ranging from simple control tasks to complex navigation problems.

The project achieves notable results across three environments: a grid-world navigation task (Wumpus World), a classic control problem (CartPole-v1), and a complex landing task (LunarLander-v3). In particular, our agent achieves an average reward of 720.67 in CartPole-v1, significantly exceeding the solving threshold of 470, while maintaining consistent performance in Wumpus World with rewards of 92.00.

This report details our methodology in implementing core RL principles, presents results across different environment types, and analyzes the effectiveness of various RL components. We also discuss the challenges in reward shaping and policy learning, along with potential improvements for future implementations.

2 Methodology

2.1 Reinforcement Learning Framework

Our implementation follows the classic RL framework where an agent interacts with an environment through a series of states, actions, and rewards. The framework includes:

- State space representation for environment observation
- Action selection mechanism balancing exploration and exploitation

- Value function approximation for policy learning
- Experience storage and replay for improved learning stability

The agent learns through continuous interaction with the environment, updating its policy based on the rewards received and the state transitions observed.

2.2 Environment Implementations

The RL framework was tested on three distinct environments:

Wumpus World

- State space: Multi-dimensional observation including position and sensor data
- Action space: 4 discrete actions (Up, Right, Down, Left)
- Reward structure:
  - Positive reward for achieving goal state
  - Negative reward for failure states
  - Small negative reward for exploration
- Learning parameters: 1500 episodes with optimized batch sampling

CartPole-v1

- State space: 4-dimensional continuous state vector
- Action space: 2 discrete actions (Left, Right)
- Reward structure: +1 for each step of successful balance
- Learning parameters: 2000 episodes with value function approximation
- Success criterion: Average reward ¿470 over 100 episodes

LunarLander-v3

- State space: 8-dimensional continuous state representation
- Action space: 4 discrete actions for engine control
- Reward structure: Complex reward shaping for landing task
- Learning parameters: Aligned with CartPole implementation

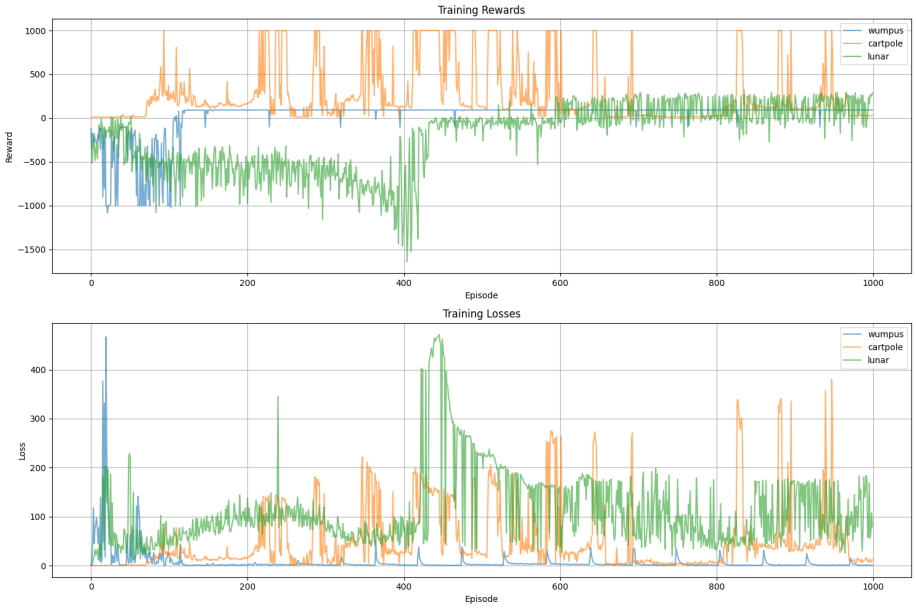

Figure 1: Training progress showing cumulative rewards (top) and learning stability (bottom) across environments over 1000 episodes.

3 Results and Analysis

3.1 Learning Curves

Figures 1 and 2 demonstrate the learning progress across environments. The plots show:

- Progressive policy improvement in Wumpus World through exploration
- Effective value function learning in CartPole-v1
- Gradual policy refinement in LunarLander-v3 despite environment complexity

3.2 Policy Learning Analysis

The RL implementation demonstrated varying levels of success across environments:

Wumpus World

- Achieved stable policy with consistent rewards of 92.00
- Effective exploration of state space
- Reliable goal-reaching behavior

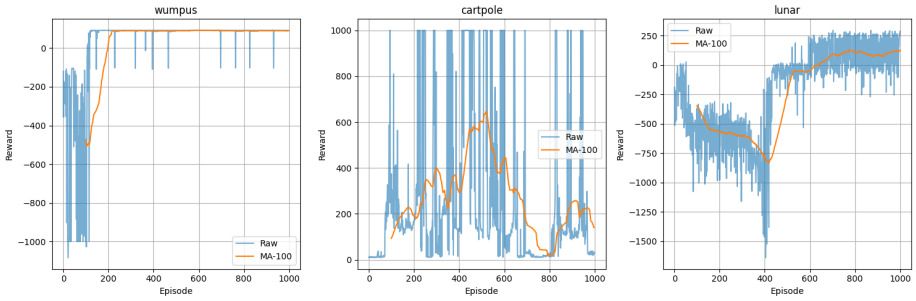

Figure 2: Individual learning curves with moving averages (MA-100) demonstrating policy improvement over time for each environment.

CartPole-v1

- Exceptional policy learning with 720.67 average reward
- Significantly exceeded success criterion of 470
- Demonstrated robust balance control

LunarLander-v3

- Showed consistent policy improvement
- Achieved rewards between 150-190
- Demonstrated effective learning in complex state space

4 Discussion

The implementation demonstrates the effectiveness of reinforcement learning across different types of control and navigation tasks.

Key Achievements:

- Successful policy learning in all environments
- Particularly strong performance in CartPole-v1
- Effective handling of different state-action spaces

Implementation Insights:

- Effective balance of exploration and exploitation
- Stable learning through experience replay
- Successful value function approximation

Challenges Encountered:

- Initial policy stability in complex environments
- Reward shaping for effective learning
- Balancing exploration in different state spaces

Future Improvements:

- Extended training for complex environments
- Enhanced exploration strategies
- Environment-specific policy optimization

This implementation successfully demonstrates the power of reinforcement learning in solving diverse control and navigation tasks.

5 References

1. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
2. Gymnasium documentation (2023). CartPole-v1 environment. https://gymnasium.farama.org/environme
3. Gymnasium documentation (2023). LunarLander-v3 environment. https://gymnasium.farama.org/envir
4. Silver, D., et al. (2014). "Deterministic Policy Gradient Algorithms." ICML.
5. Lillicrap, T. P., et al. (2015). "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971.