Project Proposal: Applying Reinforcement Learning to the Game of 2048

Jiasheng Ni, Haohai Pang

March 31st, 2025

1 Introduction

This project proposal outlines an investigation into the application of Reinforcement Learning (RL) algorithms to the classic puzzle game **2048**. The objective is to explore and compare various RL methods in the context of this game, with a focus on understanding the trade-offs between model-free, value-based, and model-based approaches.

2048 is a single-player, turn-based puzzle game played on a 4×4 grid. The goal is to combine numbered tiles by sliding them in one of four directions (up, down, left, or right) to create a tile with the value of 2048 or higher. When two tiles of the same value collide, they merge into a single tile of double the value. After each move, a new tile (either 2 or 4) appears in a random empty cell. The game terminates when no valid moves remain.

2 Research Problem

The primary objective of this project is to define and implement key RL components for the game of 2048, and to evaluate the performance of different RL algorithms in achieving high scores and long-term gameplay efficiency. Specifically, we aim to address the following questions:

- How can we effectively model the 2048 environment as an RL problem?
- Which RL methods yield the best performance in this context?
- What are the computational costs and trade-offs associated with different RL approaches?

3 RL Problem Formulation

The RL framework for 2048 is formalized as follows:

3.1 Environment ($\mathcal{E}$)

The environment corresponds to the dynamics of the 2048 game, represented by a 4×4 grid. It enforces the game's movement and merging rules, as well as the random placement of new tiles after each move. The episode ends when no valid actions are possible.

3.2 State Space ($s \in \mathcal{S}$)

The state s is defined by the current configuration of the board:

$$s = \begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{bmatrix} \tag{1}$$

where $x_{i,j}$ denotes the tile value at position (i, j). For learning stability, tile values may be normalized using $\log_2(x_{i,j} + 1)$.

3.3 Action Space ($a \in \mathcal{A}$)

The agent can choose from four discrete actions:

$$\mathcal{A} = \{\text{up}, \text{down}, \text{left}, \text{right}\} \tag{2}$$

3.4 Reward Function (r)

The reward at each time step is defined as the sum of the values of the merged tiles:

$$r_t = \sum (\text{merged tile values at step } t) \tag{3}$$

Alternative reward schemes may include bonus rewards for achieving target tiles, penalties for invalid moves, or rewards based on overall score improvement.

3.5 Policy (π)

The policy π defines the agent's action-selection strategy:

$$\pi(a|s) = \Pr(a \mid s) \tag{4}$$

In value-based methods such as Q-learning, the policy follows an ϵ-greedy strategy based on $Q(s, a)$. In policy-gradient methods such as PPO, the policy is directly parameterized by a neural network.

4 Proposed Methods

To comprehensively evaluate RL approaches in 2048, we propose to implement and compare the following algorithms:

- **Model-Free Method**: Proximal Policy Optimization (PPO), a policy-gradient technique.
- **Value-Based Method**: Deep Q-Network (DQN), which learns an approximation of the state-action value function $Q(s, a)$.
- **Model-Based Method**: A simple Expectimax-based agent that simulates tile spawn dynamics and evaluates potential future board states.

5 Evaluation Criteria

The following criteria will be used to evaluate the effectiveness and efficiency of each RL method:

- **Effectiveness**: The agent's ability to consistently reach the 2048 tile and beyond.
- **Computational Cost**: The number of training episodes and computational resources required to achieve satisfactory performance.
- **Performance Metrics**: Average game score, highest tile achieved, and the stability of the learned policy across multiple runs.

6 Expected Contributions

Through this project, we aim to provide insights into the applicability and performance of different RL strategies in a non-trivial puzzle environment. Our comparison will help elucidate the trade-offs between model-free, model-based, value-based, and policy-based RL approaches in the context of stochastic, deterministic, and combinatorial environments like 2048.