RLCard: A Platform for Reinforcement Learning in Card Games

Daochen Zha\textsuperscript{1*}, Kwei-Herng Lai\textsuperscript{1*}, Songyi Huang\textsuperscript{2*}, Yuanpu Cao\textsuperscript{1}, Keerthana Reddy\textsuperscript{1}, Juan Vargas\textsuperscript{1}, Alex Nguyen\textsuperscript{1}, Ruzhe Wei\textsuperscript{1}, Junyu Guo\textsuperscript{1} and Xia Hu\textsuperscript{1}

\textsuperscript{1}Department of Computer Science and Engineering, Texas A&M University, College Station, USA
\textsuperscript{2}Simon Fraser University, BC, Canada
\{daochen.zha, khlai037, keerthana96, juancvg, nguyen.alex, guojunyu, xiahu\}@tamu.edu, songyih@sfu.ca, yuanpucao@gmail.com, ruzhe.wei@outlook.com

Abstract

We present RLCard, a Python platform for reinforcement learning research and development in card games. RLCard supports various card environments and several baseline algorithms with unified easy-to-use interfaces, aiming at bridging reinforcement learning and imperfect information games. The platform provides flexible configurations of state representation, action encoding, and reward design. RLCard also supports visualizations for algorithm debugging. In this demo, we showcase two representative environments and their visualizations. We conclude this demo with challenges and research opportunities brought by RLCard. A video is available on YouTube\textsuperscript{3}.

1 Introduction

Reinforcement learning (RL) is a promising paradigm towards Artificial Intelligence (AI). Through interacting with the environment, RL aims to train an agent to make sequential decisions to achieve a goal by exploiting reward signals [Sutton and Barto, 2018]. With deep neural networks as function approximators, deep reinforcement learning (DRL) has recently achieved breakthroughs in various domains, such as Atari games [Mnih et al., 2015], Go game [Silver et al., 2017], and continuous control [Lillicrap et al., 2015].

Out of these achievements, however, DRL is still immature and not ready to be applied in many real-world problems. Current DRL algorithms are often less successful when dealing with long horizons, multiple agents, large decision space or sparse reward. Addressing these challenges is essential to bringing DRL to broader applications.

Card games are ideal testbeds for advancing DRL since many games have one or more of the above challenges. For example, Texas Hold’em is played by multiple players with large decision space, where each player needs to play against the other player and reason about the other players’ cards that are hidden from her sight. Another example is a popular Chinese poker game Dou Dizhu, which suffers from long sequences, sparse reward, and large action space with an explosion of card combinations. These card games are nice abstractions of many real-world problems. Moreover, card games are easy to understand with huge popularity. We usually do not need to spend efforts on learning the rules before we can dive into algorithm development.

In this demo, we present RLCard\textsuperscript{2}, a platform designed for reinforcement learning research and development in card games. RLCard aims at providing easy-to-use interfaces and evaluation tools so that users can focus on algorithm development instead of engineering efforts on games.

2 RLCard Platform

An overview in shown in Figure 1. RLCard provides unified interfaces for seven popular card games, including Blackjack, Leduc Hold’em (a simplified Texas Hold’em game), Limit Texas Hold’em, No-Limit Texas Hold’em, UNO, Dou Dizhu and Mahjong. Moreover, RLCard supports flexible environment design with configurable state and action representations. Last but not least, RLCard provides visualization and debugging tools to help users understand their algorithms.

2.1 Interface Design

In RLCard, each agent has a local view of the game. The states are all the possible observations from the view of an agent, such as cards in hand and community cards. The actions are the legal moves of a player, such as “check” in Leduc Hold’em and “33” in Dou Dizhu.

\textsuperscript{*}Those authors contributed equally to this project.
\textsuperscript{1}https://youtu.be/krK2jmSdKZc
\textsuperscript{2}https://github.com/datamlab/rlcard

Figure 1: RLCard wraps card games with unified interfaces (left). RLCard helps users develop algorithms (right).
Figure 2: Visualization modules in RLCard of Dou Dizhu (left) and Leduc Hold’em (right) for algorithm debugging. The Control Panel provides functionalities to control the replay process, such as pausing, moving forward, moving backward and speed control. The Analysis Panel displays the top actions of the agents and the corresponding probabilities.

Listing 1: Example function calls.

```python
# Initialize the environment
evaluator = rlcard.make('doudizhu')

# Initialize random agents
agent = RandomAgent()
evaluator.set_agents([agent, agent, agent])

# Generate data from the environment
trajectories, payoffs = evaluator.run()
```

2.2 Baseline Algorithms

We implement several baseline algorithms so that users can benchmark their algorithms against these existing models. We implement two deep reinforcement learning algorithms, including DQN [Mnih et al., 2015], and Neural Fictitious Self-Play (NFSP) [Heinrich and Silver, 2016]. NFSP is a deep reinforcement algorithm adapted for imperfect information games. We also implement Counterfactual Regret Minimization (CFR) [Zinkevich et al., 2008], which searches for the best strategies upon the game tree. In addition, we implement some rule-based agents as baselines. Many of the baseline results are presented in [Zha et al., 2019a].

2.3 Evaluations and Visualizations

RLCard provides evaluation and visualization tools to help users understand their algorithms, as shown in Figure 2. Firstly, RLCard has a leader-board module, where users can easily compare a new algorithm with existing baselines, i.e., the pre-trained models and the rule baselines. Secondly, RLCard supports visualizations of replay data to enable algorithm debugging. One can analyze the top actions of the agents with the visualization modules to understand the strengths and weaknesses of the agents. The users can also put their trained models into RLCard as baselines.

3 Conclusions

Card games are nice abstractions of many real-world problems. RLCard brings many challenges, such as long horizons, multiple agents, large decision space and sparse reward, and also research opportunities for tackling these challenges. In the future, we plan to incorporate more recent RL techniques, such as PPO [Schulman et al., 2017], SAC [Haarnoja et al., 2018], and better replay buffer [Zha et al., 2019b]. We will also support more card games, such as Gin Rummy.
Acknowledgements
We would like to thank JJ World Network Technology Co., LTD for the generous support, Chieh-An Tsai for user interface design, and Lei Pan for the help in visualizations.

References


3https://anntsai.myportfolio.com/
4lpa25@sfu.ca