An AI for Dominion using Deep Reinforcement Learning

Eric Yang (iame ric@stanford.edu), Yu-Chi Kuo (yu chikuo@stanford.edu)

Introduction (Problem statement)

Rules for the game Dominion
- Dominion is a deck building card game for 2-4 players. Player with the most victory points wins.
- Victory point cards are added during the game.
- Sequential planning for deck building.
- More info on http://wiki.dominionstrategy.com/

Challenges
- Stochastic nature of this game.
- This card game involves shuffling cards.
- Outcomes of a given action are not deterministic.
- Dynamic action set. Different cards are affordable at each step.

Goal
- Build a AI that decides what cards to buy at the buy phase of the game.
- The AI should be able to achieve good performance in a 2 player game with a fixed set of cards.

Approach: Deep Reinforcement Learning

- Reinforcement Learning is the framework for learning sequential decision making.
- Deep learning provides neural network as universal function approximator.
- Deep RL uses neural network to learn functions to improve decision making.

3 tried model-free RL algorithms
- Monte Carlo Learning: aggregates rewards over episodes
  
  \[ Q(s,a) = \sum_{t=0}^\infty \gamma^t R(s_t,a_t) \]

- SARSA: bootstrapping with target network
  
  \[ Q(s,a) = R(s,a) + \alpha Q(s',a') \]

- Deep Q-Learning: off-policy bootstrapping
  
  \[ Q(s,a) = R(s,a) + \alpha \max_{a'} Q(s',a') \]

Reward Design
- Victory points bought as reward at each step.
- A terminal win reward Rw: true reward but might be too sparse.
- A terminal points per turn reward Rp: motivates winning fast when the agent is stronger than opponent.

\[
\tau(s,a) = \begin{cases} 
\gamma R, & \text{if } s_t = s_f \text{ and win}, \\
\gamma R + R_p / N_{max}, & \text{if } s_t = s_f \text{ and lose}. 
\end{cases}
\]

Summary
- Hyperparameters and RL reward
  - Epislon-greedy exploration for RL agents
    \[
    \pi(a|s) = \begin{cases} 
    \tau(a|s) / \tau(s), & \text{if } a \text{ is a marginal Q}(s,a) \text{ is the expected reward at state } s \text{ if action } a \text{ is taken.} \\
    \text{otherwise}, & \text{otherwise} 
    \end{cases}
    \]
  - A decaying epislon balance between exploration and exploitation
  - A decaying epsilon balance between exploration and exploitation
  - A decaying epsilon balance between exploration and exploitation

Evaluation performance against heuristics AI/RL agent itself.

Results and Discussions

Comparison of RL algorithms
- Monte Carlo Learning performs the best.
- Most agents converge to high win rate within 100 iterations.
- Monte Carlo Learning can achieve good performance in a 2 player game with a fixed set of cards.

Future Work
- Generalize to more cards. The complete Dominion game consists of 12 expansions and >300 different kingdom cards whereas this project is only limited to 12 basic cards.

- Generalize the AI beyond 2 player games into a multiple player game and see how the optimal strategy changes.

- Use RL approach in action phase of the game.
- Build an AI that can play well with different combinations of kingdom cards including combinations that it has not seen before.

References
- [Online; accessed 10-October-2019]
- [Online; accessed 10-October-2019]