Deep Cue Learning: A Reinforcement Learning Agent for Pool

Peiyu Liao
pyliao@stanford.edu
Department of Computer Science

Nick Landy
nlandy@stanford.edu
Department of Electrical Engineering

Noah Katz
nkatz3@stanford.edu
Department of Electrical Engineering

nkatz565/CS229-pool

Summary

The goal of this project is to apply Reinforcement Learning to the game of pool.

The environment is formulated as an MDP and solved with Q-Table, DQN, and A3C algorithms.

With two balls on the table, Q-Table learns the best, but A3C with discrete action space has the best performance considering all trade-offs.

Problem Formulation

Markov Decision Process (MDP)

- **State** `s`: list of x, y positions for each ball, white ball first
- **Action** `a`: angle, force $\in [0, 1]$
- **Reward** `R(s, a)`: 5 for each ball pocketed
  -1 if no ball collides with white ball
  0 otherwise

Algorithms

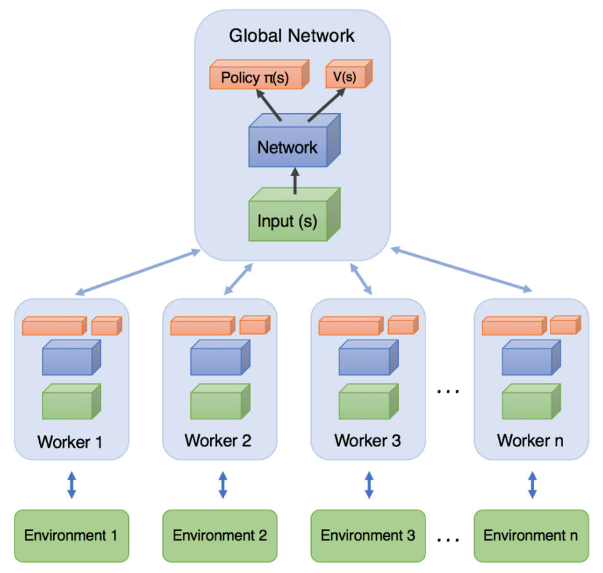
Q-Table: Implements Q-learning with discretized states and actions, uses a lookup table for each (s, a) pair to represent the Q-function.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \max_{a'} Q(s',a') - Q(s,a))$$

Deep Q-Network (DQN) [1]: Uses DNN to approximate the Q-function, with continuous state values as input and the Q-values for each discrete action as output.

Asynchronous Advantage Actor-Critic (A3C) [2]:

Estimates both the value function and policy; policy can be updated more intelligently with the value estimate. Multiple agents learn asynchronously on different threads to speed up the overall training.



Experimental Results

2 Balls (Episode Length 25)

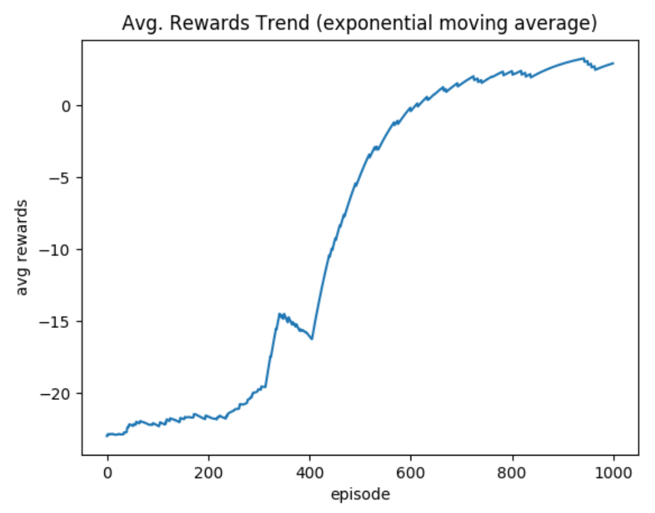
Q-Table
state: 50 buckets
angle: 18 buckets; force: 5 buckets

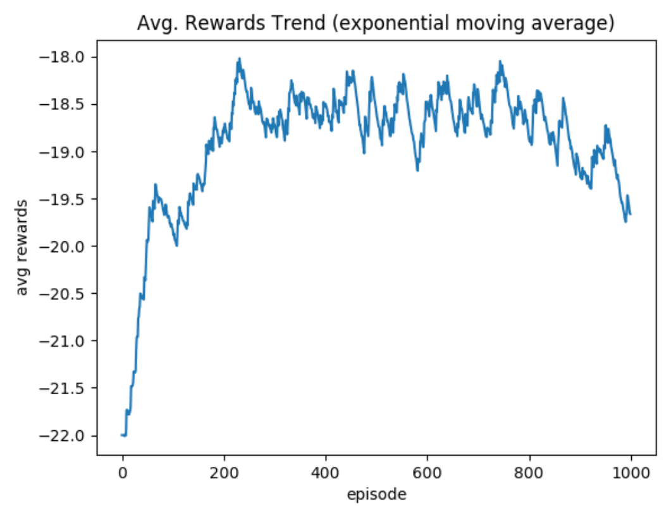
DQN

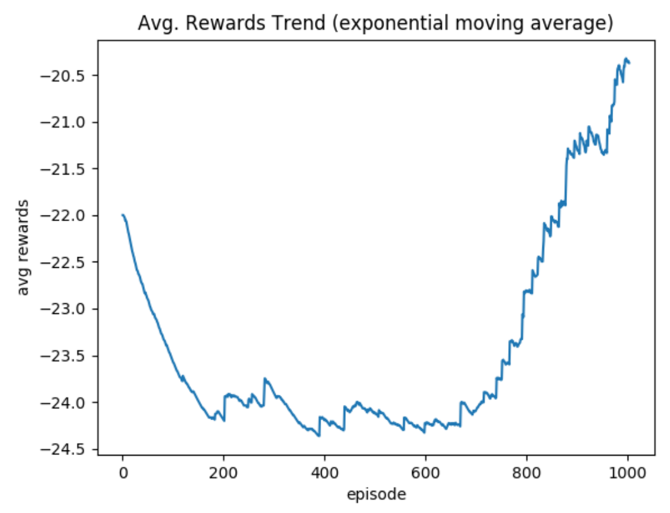
A3C (continuous action)

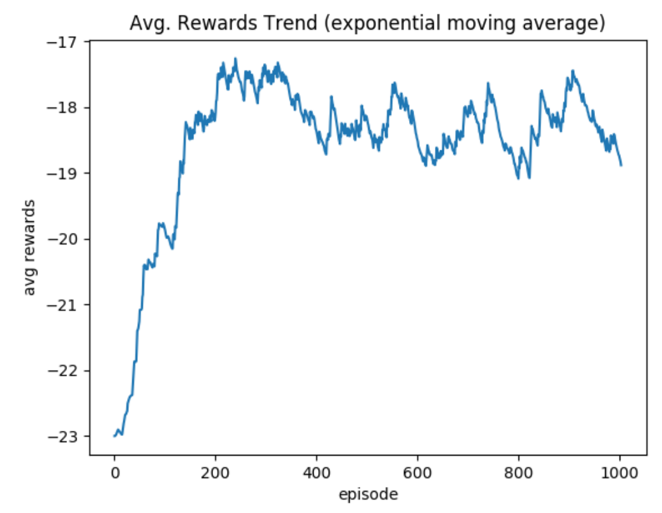
A3C (discrete action)
angle: 360 buckets; force: always max

Training Results (over 1000 episodes)





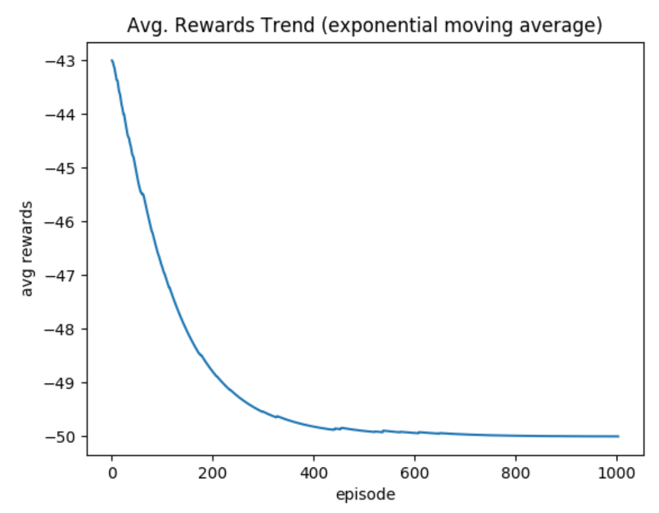


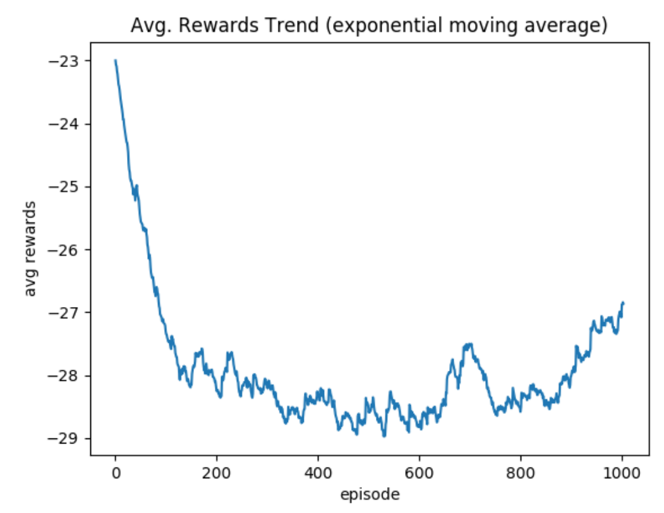


4 Balls (Episode Length 50)

A3C (continuous action)

A3C (discrete action)





Evaluation Results (over 100 episodes) & Training Statistics

	Q-Table	DQN	A3C (continuous action)	A3C (discrete action)	A3C (continuous action)	A3C (discrete action)
	2 Balls	2 Balls	2 Balls	2 Balls	4 Balls	4 Balls
Average reward:	-6.4	-21.3	-19.44	-18.46	-50.0	-24.86
Training time:	136 min	27 min	13 min	17 min	28 min	52 min
Model size:	1.12 GB	162 KB	8 KB	149 KB	11 KB	152 KB

Discussion

Two-Ball Environment

Q-Table: Best performance, learns the exact steps to hit the ball in from the starting position. Table size large, limited to two-ball environment. Training time significantly longer.

DQN: Good performance, but training unstable. Model learns only two or three good actions that tend to get better total rewards, but does not do well in the short term.

A3C (continuous action): Good performance, but longer convergence time. Space efficient, generalizable to larger environments. Since it predicts mean and variance of the normal distributions for actions, it is hard for the values to settle in the right range.

A3C (discrete action): Better performance than with continuous action. Sacrifices some accuracy, space, and time, but classification training is more effective than predicting bounded continuous values.

Four-Ball Environment

Both A3C with continuous and discrete action perform poorly when state space is enlarged.

In continuous action, values tend to be saturated and clipped at 0 or 1; in discrete action, a single angle value tends to be favored.

Future Work

Inspect the value saturation problem in A3C, look for improvements in environments with more balls.

Compete the AI with human player for more comprehensive evaluation.

References

[1] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529.

[2] Mnih, Volodymyr, et al. "Asynchronous methods for deep reinforcement learning." International conference on machine learning. 2016.