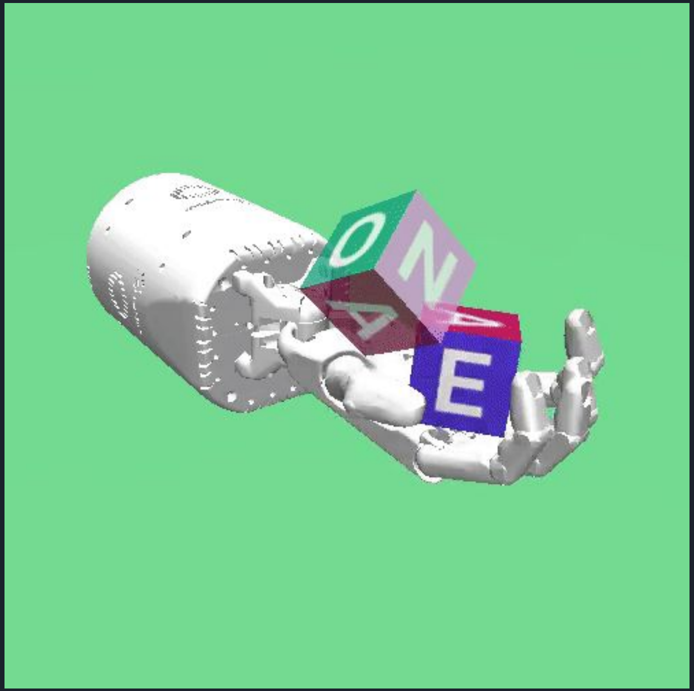
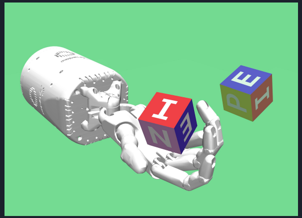
Applying Q-Learning to Continuous State-Action Spaces Using Discretization

David Tandetnik

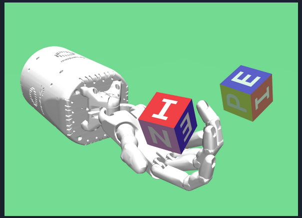
OpenAI Gym: Robot Hand Manipulate Block





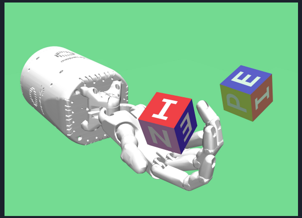
Background: OpenAI Research

- OpenAI researchers used Deep Deterministic Policy Gradient (DDPG) algorithm
  - Q-Learning for continuous state-action spaces
  - Like Q-Learning, but uses separate networks to approximate max[Q(s',a)]
- DDPG works well, but involves relatively large amounts of CPU and memory compared to basic Q-Learning
- Can we use discretization to simplify the continuous state-action space and solve with basic Q-Learning?



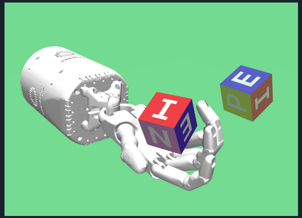
Background: OpenAI Research

- OpenAI researchers used Deep Deterministic Policy Gradient (DDPG) algorithm
  - Q-Learning for continuous state-action spaces
  - Like Q-Learning, but uses separate networks to approximate max[Q(s’,a)]
- DDPG works well, but involves relatively large amounts of CPU and memory compared to basic Q-Learning
- **Can we use discretization to simplify the continuous state-action space and solve with basic Q-Learning?**



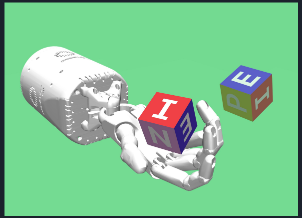
Simplifying Goal Definition

1. Target Z rotation
2. Target XY rotation
3. Target XYZ rotation
4. Target XYZ rotation and position in space



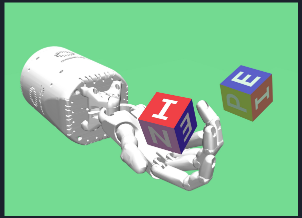
Simplifying Goal Definition

1. **Target Z rotation**
2. Target XY rotation
3. Target XYZ rotation
4. Target XYZ rotation and position in space



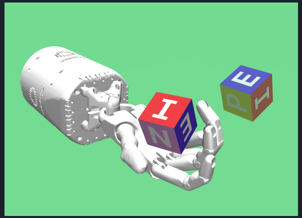
Discretization and Simplification: State Space

- Original state vector
  - [Robot joint positions ... Robot joint velocities ... Block position ... Block velocity ... Block rotation ... Target rotation]
  - 61 arbitrary floats
  - Continuous/infinite state space
- Discretized and simplified state vector
  - [Block's Z rotation ... Target Z rotation]
  - 4 floats between -1.0 and 1.0, rounded to nearest tenth
  - ~200k state space

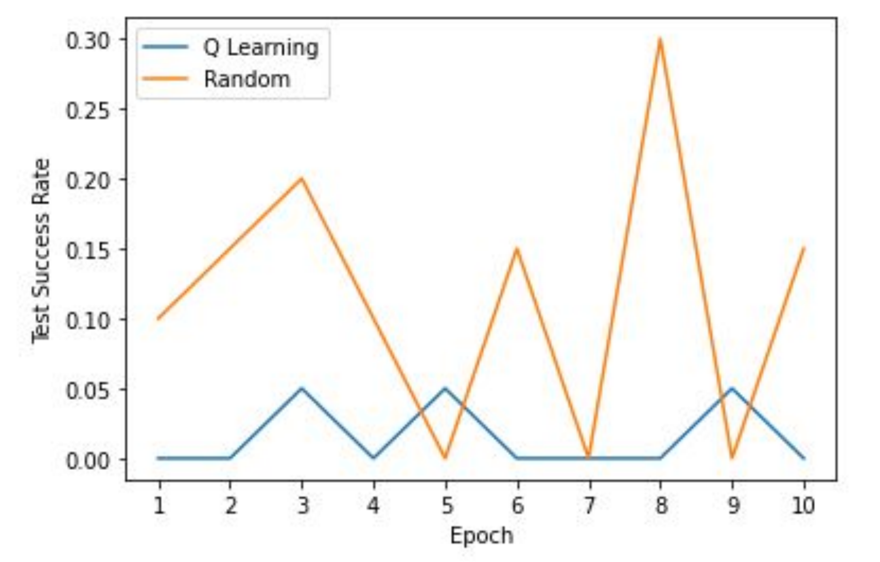


Discretization and Simplification: Action Space

- Original action vector
  - [Absolute robot joint positions]
  - 20 arbitrary floats
  - Continuous/infinite state space
- Discretized and simplified action vector
  - [Relative robot joint positions]
  - 20 floats, 10 set to 0 and 10 set to one of -0.25, 0.25
  - 1024 action space size
- We've reduced O(|state||action|) to ~10^8

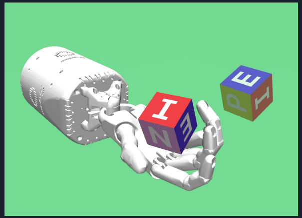


Results

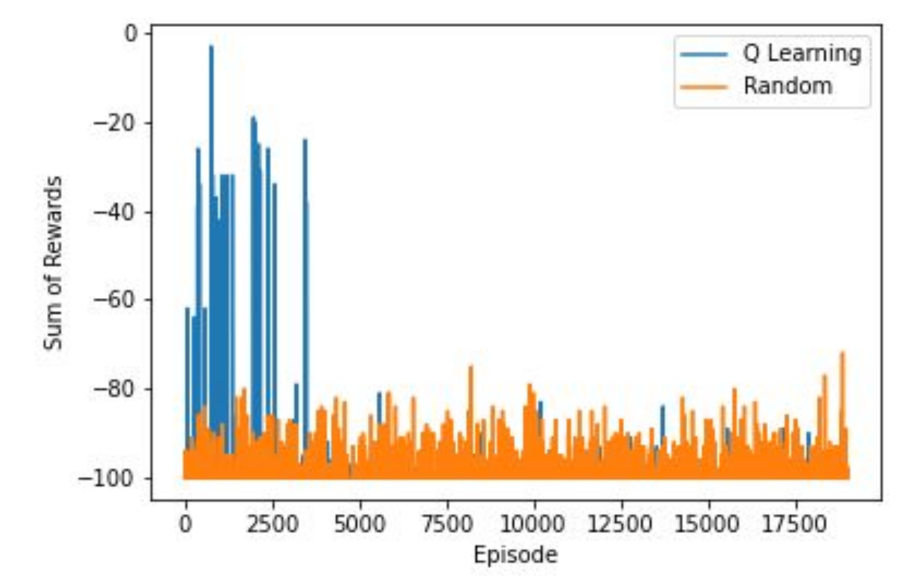


- For reference, DDPG after 10 epochs had a test success rate of about 0.2
- After 25 epochs, DDPG eventually reaches 0.8

Success rate of 20 test episodes run after each epoch. One epoch is 1900 episodes. Success defined as at least one non-negative reward during episode.

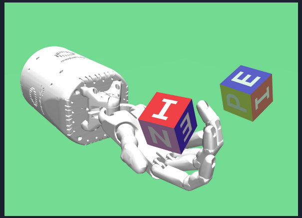


Results



Sum of rewards per episode.

- Overall not much better than random
- Some success in first couple epochs
  - Perhaps by chance some episodes had nearby start and target block orientations



Conclusions and Next Steps

- State-Action space still too large. Never gets a chance to use what it learns
  - Need to either reduce it further or increase computation time (i.e. number of episodes)
- Upcoming week
  - Experiment with introducing noise to static joint actions
  - Increase training time
  - Reduce state and/or action space further
- Questions?