Developing Simulation Environments and Applying Deep Reinforcement Learning Algorithms for the SoftGym Project

Jake Olkin
Advisor: David Held

Carnegie Mellon University
The Robotics Institute

RpAd
Robots Perceiving And Doing

Problem Statement

This thesis seeks to create a standard set of tasks for researchers to use when evaluating their approaches to manipulating deformable objects. This is part of the SoftGym project (Lin et al. 2020) to benchmark reinforcement learning algorithms on deformable objects. While some of the algorithms learned the tasks, there were systematic difficulties when training the algorithms with the TD3 algorithm (Fujimoto et. al. 2018) when using the images observation space.

The observation space is the representation of the environment the agent receives. In one observation space, the agent receives an array of the positions of objects. The TD3 algorithm was unable to learn any of these tasks when using an RGB image-based observation space.

The TD3 reinforcement learning algorithm failed due to action saturation. Action saturation is the result of an agent only performing actions that are either the absolute maximum or minimum value. The lack of variation in the movement of the agent leads to a flattening of the loss curve, which in turn results in an inability for the algorithm to learn the task.

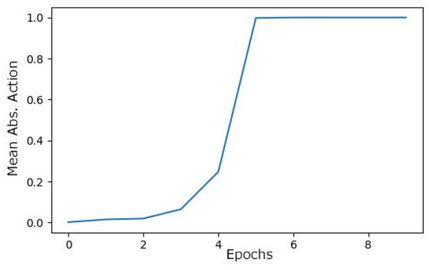


Figure 1. An example of action saturation.

After only five epochs the mean of the absolute value of the agent's actions reached the maximum value of 1.0 during this training of a policy in the Water Pouring environment using the TD3 reinforcement learning. Each epoch consisted of 120 exploration steps.

Acknowledgement

I would like to thank Prof. Held for letting me work in his lab. I would like to thank Yufei Wang and Xingyu Lin for their guidance on this project.

References

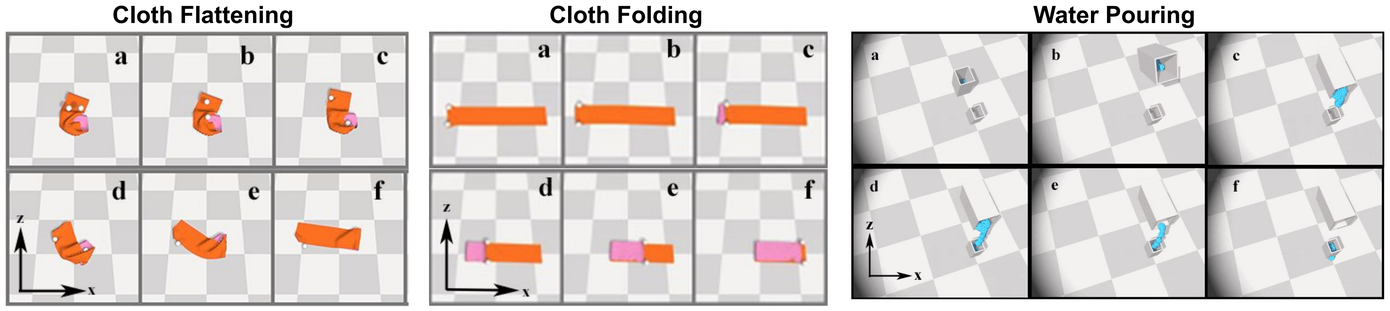
1. Lin, Wang, Olkin, and Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. Submitted to ICML20, 2020.

2. Fujimoto, van Hoof, and Meger. Addressing function approximation error in actor-critic methods. CoRR, abs/1802.09477, 2018.

Environments and Tasks

Figure 2. The three tasks used in this project.

These environments were built using NVIDIA FleX. The cloth is represented by a grid of particles.



Reward Function: Surface area covered by the cloth in the xz plane.

Reward Function: Distance between matching pairs of particles.

Reward Function: Number of water particles in the target cup.

Reinforcement Learning

The following parameters were investigated in initial attempts to alleviate the action saturation problem:

(1) the noise added to the policy during exploration (σ),
(2) the probability that the agent would take a uniformly random action (ε),
(3) the learning rate (μ),
(4) the discount and
(5) the number of exploration steps per epoch.

No combination of these parameters fixed the action saturation issue.

Test No.	σ	ϵ	μ	Discount	Steps
Baseline	0.1	0	10^{-3}	0.99	120
1	0.02	0.0	10^{-6}	0.5	120
2	0.5	0.5	10^{-6}	0.5	120
3	0.5	0.5	10^{-6}	0.25	120
4	0.1	0.0	10^{-6}	0.25	120
5	0.5	0.75	10^{-6}	0.99	120
6	0.5	0.5	10^{-6}	0.99	360
7	0.5	0.5	10^{-6}	0.99	360
8	0.5	1.0	10^{-6}	0.99	360

The key to removing action saturation in the Water Pouring task was to modify the weight decay (λ). This parameter is a penalty based on the magnitude of the weight. This changes the update equation for weight W_i in the policy to include a new weight decay term:

$$W_i = W_i - \mu \frac{dL}{dW_i} - \mu \lambda W_i$$

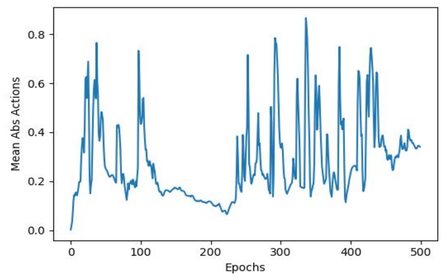


Figure 3. No action saturation for the Water Pouring task.

Throughout the entire training, the mean of the absolute value of the actions taken by the agent never reaches its maximum value unlike in the saturation case.

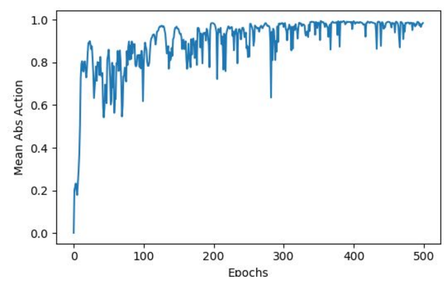


Figure 4. Action saturation for the Cloth Flattening task.

Running the TD3 algorithm with the weight decay on the cloth flattening algorithm did not alleviate the action saturation.

Even though the action saturation was eliminated for the Water Pouring task, the TD3 algorithm failed to learn the task. This means that there is another issue affecting the ability of the TD3 algorithm to learn this task. Additionally, the TD3 algorithm did not learn either of the cloth tasks.

Conclusions

The addition of weight decay successfully eliminated action saturation from the pouring water environment. However, it did not have this effect on either of the cloth-related environments. These are very different environments. Taking random actions in the water pouring environment is more likely to result in the agent experiencing some reward than in the cloth environments, since the water pouring environment is always interacting with the water, and also has a smaller dimensionality action space. In contrast the cloth environments have to first maneuver the grippers to the cloth before being able to interact with it.

In addition to the higher dimensionality action spaces of the cloth tasks, there is a longer, less likely series of actions that must be taken for the agent to experience any reward from the cloth environment than in the pouring water environment. The time that it takes to touch the cloth is likely too long compared to the time it takes the pouring water environment to get any amount of water in the cup. This means that, by the time the cloth environments receive any feedback from the environment, the policies have already saturated, and cannot learn from the new information.