Optimize Robots Physical Design by Parameterization in Deep Reinforcement Learning

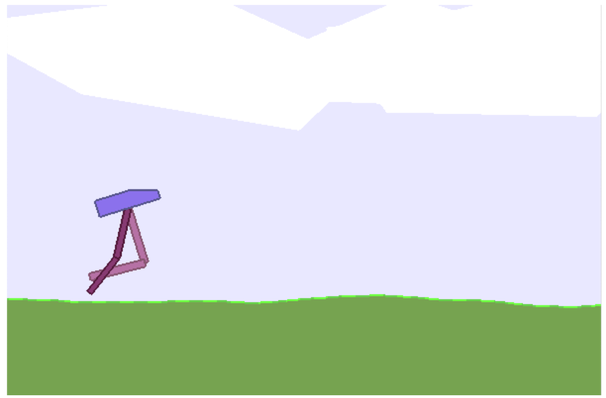
Video: https://youtu.be/9l7chjjsQr4
Peide Huang
pdhuang@stanford.edu

Motivation

- When doing mechanical design, it is difficult to determine the optimal mechanical properties for the specific task.
- When doing control, people sometimes use RL to find the optimal control policy.
- Can we enable the agent to modify some environment parameters so that it can evolve to a better version?
- This could help us find the optimal mechanical properties such as the center of gravity and the stroke of actuator when doing mechanical design.

Environment

- OpenAI Gym BipedalWalker-v2 [1]
- Observation: 24 sensor readings
- Action: torque of 4 joints

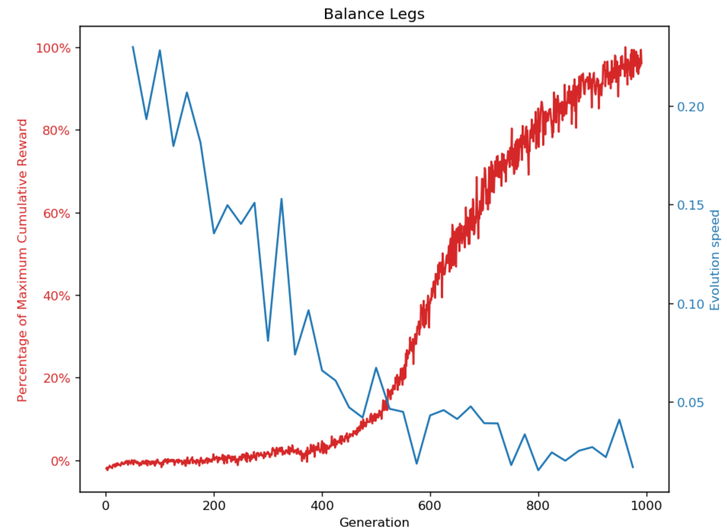


Method

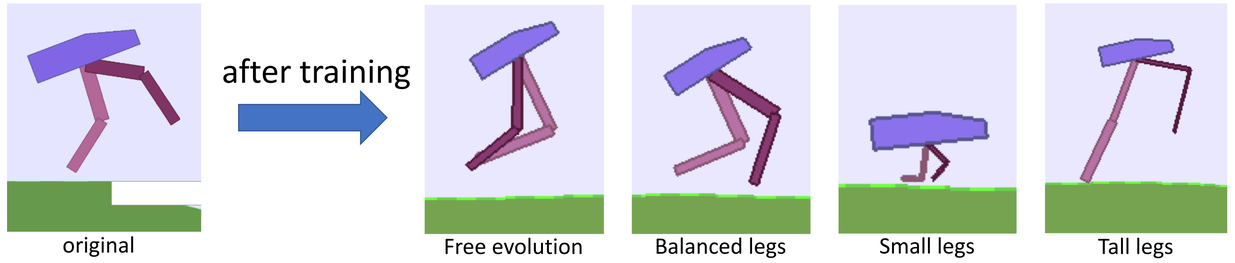
- Use two hidden layer fully connected NN to learn the optimal control policy.
- Use Population-based Policy Gradient Method to update environment parameters. [2]
- Augment reward function to force the evolution to the direction we want. For example, if we want small legs, we can factor the reward by $1.0 + \log(\frac{original\ leg\ area}{current\ leg\ area})$

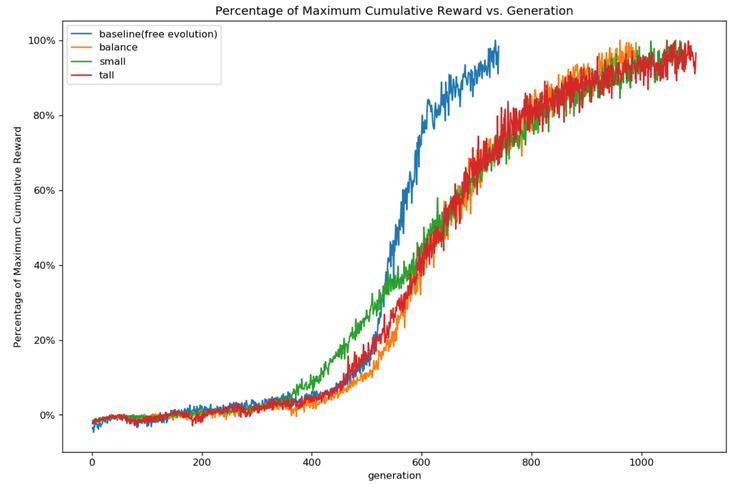
Discussion

- Constraining the evolution make the learning process less efficient than free evolution.
- The agent tries to evolve to a better version before it can start to learn the optimal control policy efficiently, as demonstrated in the figure on the right.



Results





- Trained on 96-CPU AWS machine. Training takes about 8 hours.
- The agent evolves to certain configuration as expected.
- All of the agents are able to solve the task (reach the finishing point).

Future

- Try more sophisticated neural network architectures to accelerate the learning process.
- Use different optimizers to update the environment parameters to improve the stability and convergence speed.
- Explore the possible generalization to tasks other than the bipedal walker.

Reference

[1] A toolkit for developing and comparing reinforcement learning algorithms. https://gym.openai.com/docs/. Accessed: 2010-09-30
[2] David Ha. Reinforcement learning for improving agent design. arXiv preprint arXiv:1810.03779, 2018