Continuous Control Project

September 6, 2024

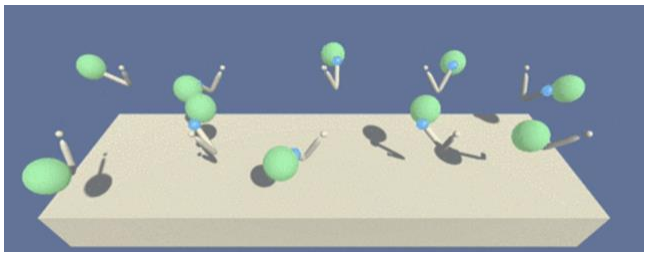
By Haya Alhuraib

---

1. Project Overview

- ## Environment

In this project, an agent or multiple similar agents is tasked with tracking a target. The agent receives a reward of +1 for each time steps it successfully maintains its hand in the goal location. The ultimate objective of the agent is to sustain its position at the target location for the maximum number of time steps possible. The environment space is defined by 33 variables by agent (position, rotation, velocity, and angular velocities of the arm) and the action space contains 4 numbers corresponding to torque applicable to two joints.



2. Learning Algorithm

The Deep Deterministic Policy Gradient (DDPG) algorithm is employed in this project. The DDPG architecture consists of two neural networks: an actor and a critic.

- **Actor-Critic Mechanism**

During each step, the actor network estimates the optimal action by computing **argmaxaQ(sa)**. The critic network then utilizes this value to evaluate the optimal action-value function, similar to a DDQN.

- **Network Architecture**

Both the actor and critic networks comprise two components: a local network and a target network. This design is necessary to avoid computational difficulties during backpropagation, where the same model would be used to compute both the target value and the prediction.

- **Training Process**

The actor is updated by applying the chain rule to the expected return from the start distribution. The critic is updated using Q-learning principles, where the expected return of the current state is compared to the sum of the reward of the chosen action and the expected return of the next state.

- **Implementation Details**

First architecture: The step function was adapted to accommodate 20 simultaneous agents returning experiences. Additionally, a unique noise was applied to each agent, as initially using the same noise for all agents resulted in unsuccessful training.

The actor is composed of 3 fc units:

First layer: input size = 33 and output size = 128

Second layer: input size = 128 and output size = 128

Third layer: input size = 128 and output size = 4

The critic is composed of 3 fc units:

First layer: input size = 33 and output size = 128

Second layer: input size = 134 and output size = 128

Third layer: input size = 128 and output size = 1

The second layer takes as input the output of the first layer concatenated with the chosen actions. The training hyperparameters are as follow:

Buffer size: 100,000

Batch size: 128

learning rate actor: 0.001

learning rate critic: 0.001

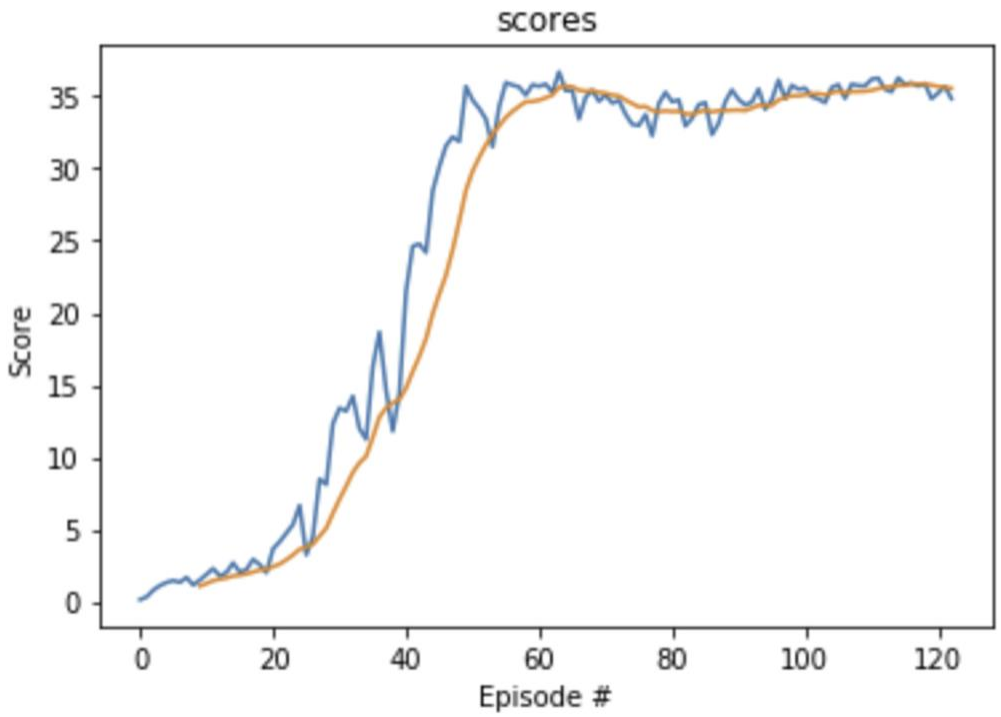
weight decay: 0

3. **Plot of rewards**

The environment has been solved in 23 episodes.

```
Episode 100     Average Score: 22.49
Episode 123     Average Score: 30.21
Environment solved in 23 episodes!      Average Score: 30.21
```

Below is the graph of the score evolution:



4. **Ideas of future works**

To enhance the stability of the model, incorporating batch normalization could be a viable option. Batch normalization facilitates a better gradient flow by ensuring that the input distribution to each layer during training consistently follows a Gaussian distribution.

Alternative Models

Additionally, exploring other models, such as those discussed in the course, could be beneficial. These include:

- Proximal Policy Optimization (PPO)

- Asynchronous Advantage Actor-Critic (A3C)
- Distributed Distributional Deep Deterministic Policy Gradients (D4PG)

These alternative models could potentially offer improved performance and stability in the project.