Ehsan Arabnezhad

Environment

The state space has 33 dimensions each of which is a continuous variable. It includes position, rotation, velocity, and angular velocities o the agent.

The action space is a subset of R^4. That is, each action is a vector of four real numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number in the interval $[-1, 1]$.

A reward of +0.1 is provided for each step that agent's hand is in the goal location. The goal of the agent is to maintain contact with the target location for as many time steps as possible.

Distributed training

For this project, 2 environments are provided:

- The first version contains a single agent
- The second version contains 20 identical agents, each with its own copy of the environment. This version is particularly useful for algorithms like PPO, A3C, and D4PG that use multiple (non-interacting parallel) copies of the same agent to distribute the task of gathering experience. Solving the environment Solve the First Version The task is episodic, and in order to solve the environment, the agent must get an average score of +30 over 100 consecutive episodes. Solve the Second Version Since there are more than 1 agents, we must achieve an average score of +30 (over 100 consecutive episodes, and over all agents).

Algorithm

Deep Deterministic Policy Gradient (DDPG) is an algorithm which concurrently learns a Q-function and a policy. It uses off-policy data and the Bellman equation to learn the Q-function, and uses the Q-function to learn the policy.

DDPG interleaves learning an approximate to Q^* (s, a) (Critic) with learning an approximate to a^*(s)(Actor), and it does so in way which is specifically adapted for environments with continuous action spaces.

Actor and Critic Network Architecture

Both the Actor and the Critic networks have 2 hidden layers each. In both networks, the 1st layer has 400 neurons while the 2nd layer has 300 neurons. Also, both networks have a batch normalization layer prior to the 1st hidden layers. The final layer in Actor Network is a linear layer with neurons (= #actions) and tanh activation. Critic network has only one neuron in its final layer with no activation. Since, the critic network estimates Q-values for all state-action pairs, we have to somehow insert the corresponding actions at some layer. We do this in the 2nd hidden layer of the Critic network.

Solution

The environment contains 20 agents working in parallel. The agents learnt from the experience tuples every 20 time steps and at very update step, the agents learnt 10 times. Gradient clipping helped improved the training. Also, to add a bit of exploration while choosing actions, Ornstein-Uhlenbeck process was used to add noise to the chosen actions.

Hyper-parameters

- Learning rate (Actor): $1e^{-3}$

- Learning rate (Critic): $1e^{-3}$

- Batch size: 1024

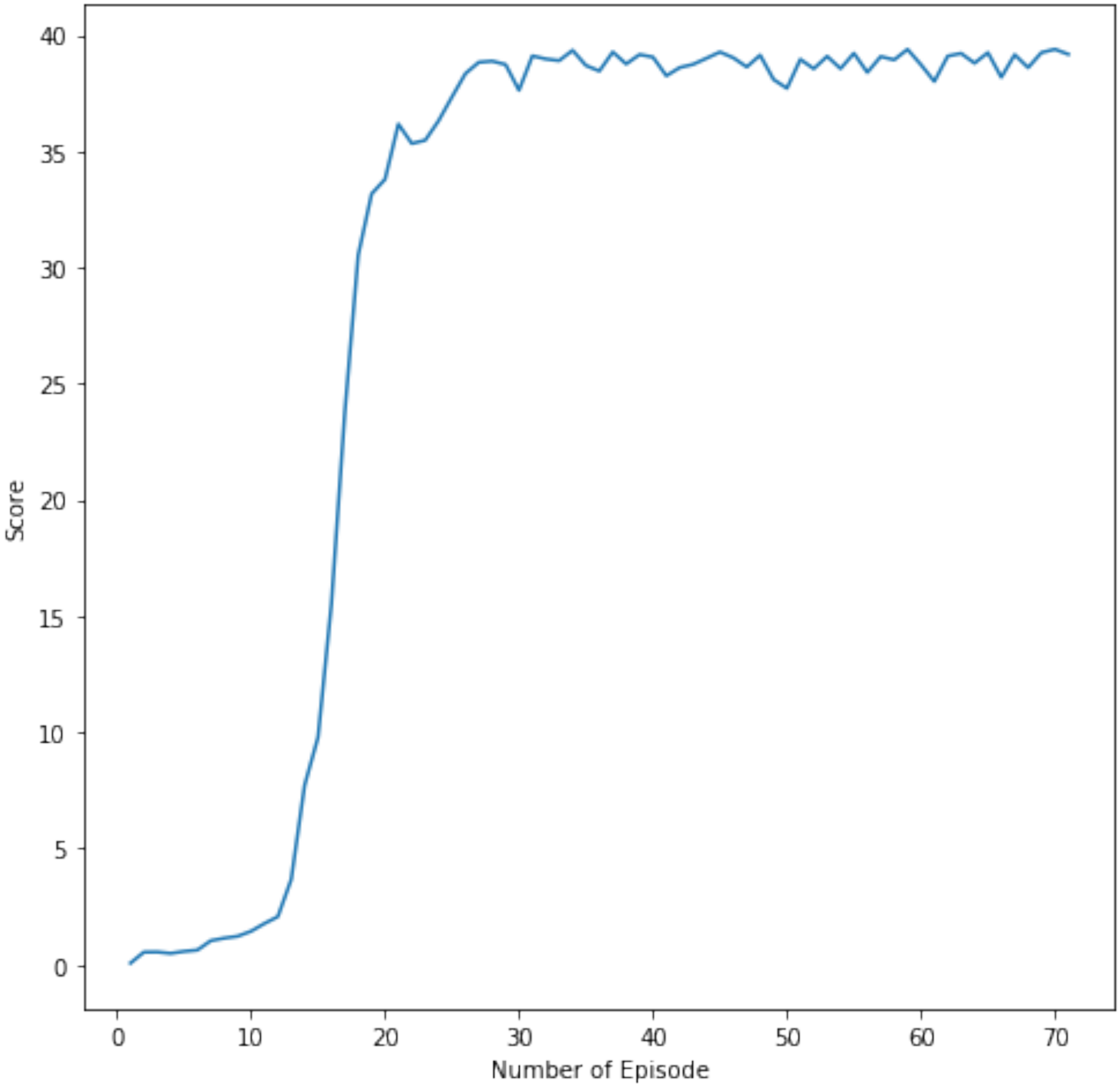
- Replay Buffer size: $1e^{6}$

- Discount factor (γ): 0.99

- Soft Update parameter (τ): 0.001

Results

The agent solved the problem in 71 episodes to achieve an average reward of 30 scores in GPU instance. A sharp increase of score is observed between 10th and 20th episode.



Future work

- Use of Prioritized Experience Replay Buffer

- T implement Trust Region Policy Optimization (TRPO) and Truncated Natural Policy Gradient (TNPG)

- To implement Distributed Distributional Deterministic Policy Gradients (D4PG)