PROJECT REPORT

Topic – Continuous Control Using Deep Reinforcement Learning
Name – Pranav Sivadas Menon

Environment –Reacher Environment from Unity Machine Learning Agents toolkit.

Goal – Train a double-jointed arm agent to follow a target location

Reward structure - A reward of +0.1 is provided for each step that the agent's hands is in the goal location. Thus, the goal of the agent is to maintain its position at the target location for as many steps as possible.

State space – The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm.

Action space - Each action is a vector with four numbers, corresponding to torque applicable to two joints. Every entry in the action vector should be a number between -1 and 1.

Stopping Criterion - The task is episodic. In order to solve the environment, the agent must get an average score of +30 over 100 consecutive episodes.

In this project I trained a single model: Parallelized version of the TD3 algorithm. It is an extension of the DDPG algorithm.

To learn more about these topics please refer the following papers:

- **Continuous control with deep reinforcement learning**
- **Addressing Function Approximation Error in Actor-Critic Methods**

Neural Network:

Actor:

- 3 linear layers. The first hidden layer has 256 units and second hidden layer has 128 units. Parameters are reset according to number of units in the layers i.e
  lim = 1. / np.sqrt(f_in)
- Output size – action_size (vector of 4 numbers since it is continuous action space).Tanh activation

Critic:

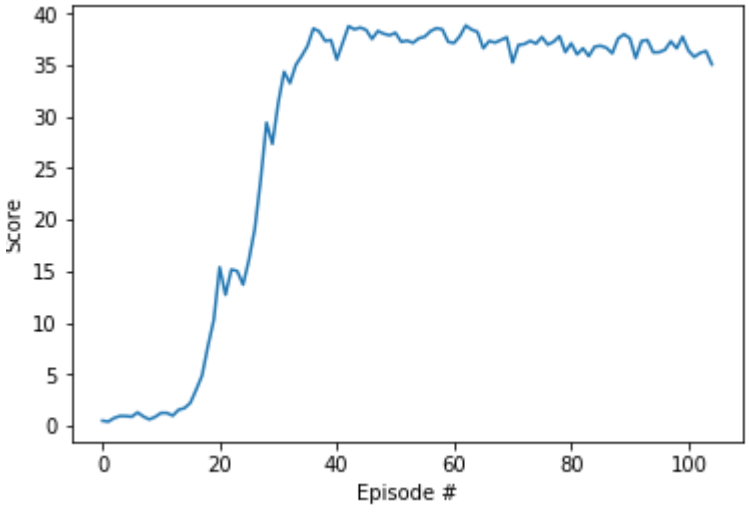
- 3 linear layers. The first hidden layer has 256 units and second hidden layer has 128 units. Parameters are reset according to number of units in the layers i.e
  lim = 1. / np.sqrt(f_in)
- The action is fed along with state as input
- Output size – 1 (action value function)

All hidden layers use Relu non linearity.

Hyperparameters chosen:

batch_size = 128
gamma = 0.99
buffer_size = int(1e5)
TAU = 1e-3
lr_actor = lr_critic1 = lr_critic2 = 1e-3
noise_clip = 0.5
start_length = 128
exploration_noise = 0.1
policy_noise = 0.2
policy_delay = 2

Plot of Rewards – Version 2



Future work:

- Try PPO,D4PG and A3C.
- Solve crawler environment