Report

Introduction

This document contains a brief report of Unity Reacher problem.

Problem Description

In this environment, a double-jointed arm can move to target locations. A reward of +0.1 is provided for each step that the agent's hand is in the goal location. Thus, the goal of the agent is to maintain its position at the target location for as many time steps as possible.

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

The goal is to reach score of +30 over 100 consecutive episodes.

Two separate versions of the Unity environment are available

- The first version contains a single agent.
- The second version contains 20 identical agents, each with its own copy of the environment.

In this report the first version has been used.

Method used

To solve this problem the DDPG algorithm is used which is an actor-critic method. The network architecture for actor and critic are as below.

Table 1 Actor Network

Layer	Activation Function	Size
Input	Relu	33
Fully connected 1	Relu	128
Fully connected 2	Tanh	128
Output	N/A	4

Table 2 Critic Network

Layer	Activation Function	Size
Input	Relu	37
Fully connected 1	Relu	128
Fully connected 2	N/A	128
Output	N/A	1

For the actor network, the size of input layer and output put layer matches the state space size and action space size respectively. For the critic network, the size of input matches the state space size plus the action size and the output layer has a size of 1.

Hyper-Parameters

The table below describes the hyper-parameters used.

Table 3 Hyper parameters

Parameter	Value
Batch size	128
Learning Rate	0.99
Gamma	0.0002
Tau	0.001

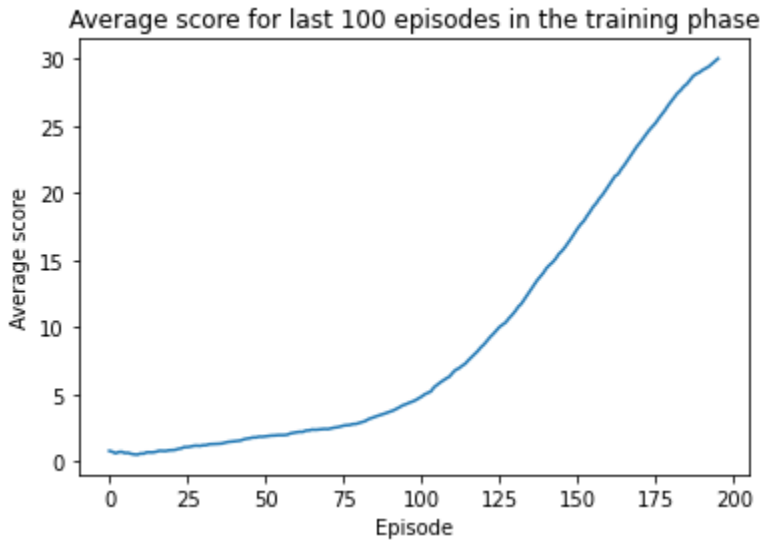
Results

The following result shows the training progress for a random run of the code.

```
Episode 100     Average Score: 4.69

Episode 196     Average Score: 30.04

Criteria reached after 196 episodes
```



Ideas for future work

Although the current results show that the model is working well, there might be a few solutions to make the agent work even better or learn faster. I suggest trying the following ideas.

- Changing the network structure
    - Adding extra layers
    - Change layers size
- Try changing the hyper-parameters
- Try using the second version of the environment