Project2_Continuous_Control

Project Objective

The goal of this project is to create and train a double-jointed arm agent that is able to maintain its hand in contact with a moving target for as many time steps as possible.

Environment:
This environment provided using the Unity ML-Agents. 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 target location.

State/Action Spaces:
The observation space consists of 33 variables corresponding to position, rotation, velocity, and angular velocities of the arm.
the actiona space consists of 4 actions

Instructions to download for (version 1 or version 2):

1. Download the environment from one of the links below:
    - ***Version 1: One (1) Agent***
        - Linux: click here
        - Mac OSX: click here
        - Windows (32-bit): click here
        - Windows (64-bit): click here
    - ***Version 2: Twenty (20) Agents***
        - Linux: click here
        - Mac OSX: click here
        - Windows (32-bit): click here
        - Windows (64-bit): click here

2. Place the file in the DRLND GitHub repository, in the p2_continuous-control/ folder, and unzip (or decompress) the file.

To set up your python environment to run the code in this repository, follow the instructions below.

1. Create (and activate) a new environment with Python 3.6.

    - **Linux** or **Mac**:

```
conda create --name drlnd python=3.6
source activate drlnd
```

    - **Windows**:

```
conda create --name drlnd python=3.6
activate drlnd
```

3. Clone the repository (if you haven't already!), and navigate to the python/ folder. Then, install several dependencies.

```
git clone https://github.com/udacity/deep-reinforcement-learning.git
cd deep-reinforcement-learning/python
pip install .
```

3. On Sucessfully installing the necessary packages and dependencies Launch the Project by navigating to the file p1_continuous-control/ where the Continuous Control.ipnb notebook file is visiable. Run the following cammand to launch the Jupyter environment to execute the notebook

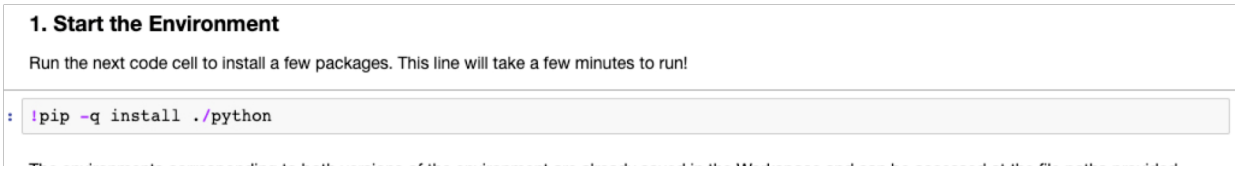
```
jupyter notebook
```

open the Single-Agent Continious Control.ipnb notebook file to train the agent .

Please download the github repository and use place all the files in a folder and open .ipynb file

https://github.com/sriten15/Project2_Continuous_Control

ipynb will install the python environment



Train and Test the agent:

Firstly, we shd import the following modules,

```
.2]: from unityagents import UnityEnvironment
     import numpy as np
     import time
     from collections import deque
     import os
     import matplotlib.pyplot as plt
     import model
     import ddpg_agent
     import torch
     from ddpg_agent import Agent
     # select this option to load version 1 (with a single agent) of the environment
     env = UnityEnvironment(file_name='/data/Reacher_One_Linux_NoVis/Reacher_One_Linux_NoVis.x86_64')
```

sriten15/Project2_Continuous_Control

Create the agent and call the training function:

```
:  # Create the Agent
   agent = Agent(state_size, action_size, random_seed=0)

   # Train the Agent
   scores, moving_avg = ddpg()
```

Checkpoints to test:
Once the agent is trained, the checkpoint.pth will hold the training parameters which can be loaded to test the agent

```
    print('\nSaving model ...', end=' ')
    torch.save(checkpoint, 'checkpoint.pth')
    print('done.')
```

Goal:

To solve the environment, one agent version , the agent is expected to achieve average score of 30+ for 100 consecutive episodes.

Github Repository Files:
Readme.md
Continuous_Control.ipynb
ddpg_agent.py model.py
checkpoint.pth
Report.md

Since the project is executed in jupyter notebook. Please run the jupyter notebook as is to look at the agent trained in the environment.