Advice-based Conservative Q-Learning

This repository contains the original implementation for *No Imitation for Me: Advice-based Offline Reinforcement Learning for Aerial Control*. Code is based on the Gym-Pybullet-Drones framework.

Requirements and Installation

Our implementation is written using *Python 3.7* and tested on *Ubuntu 18.04* using *PyTorch*. Use the following command to setup the required dependencies-

```
setup.sh

pip3 install --upgrade numpy Pillow matplotlib cycler
pip3 install --upgrade gym pybullet stable_baselines3 'ray[rllib]'
sudo apt install ffmpeg
```

Following are the installation instructions-

Major dependencies are `gym`, `pybullet`, `stable-baselines3`, and `rllib`

```
pip3 install --upgrade numpy Pillow matplotlib cycler
pip3 install --upgrade gym pybullet stable_baselines3 'ray[rllib]'
sudo apt install ffmpeg
```

The repo is structured as a Gym Environment and can be installed with `pip install --editable`

```
git clone https://github.com/utiasDSL/gym-pybullet-drones.git
cd gym-pybullet-drones/
pip3 install -e .
```

Usage

Our project is implemented using the `singleagent.py` file. This is a common file which will run all our implementations upon integration.

To run an `SAC` agent on the `takeoff` task with `rgb` images use the following-

```
python singleagent.py --configs SAC --env takeoff --obs rgb
```

This will train the agent for `1e5` timesteps and save results in the `results` folder.

The default settings will train a `SAC` agent on the `hover` task with `kin` feature inputs as per the following-

```
python singleagent.py
```

Custom implementations can be trained using config files in their respective directories in the `config` folder.

Development

So what is a good place to start your work? Look at the following-

- Code study- I would suggest you to study and understand the `learning` package before making any changes.
- `algos`- Follow a similar line of coding as in the `algos` folder as this will lead to easier integration and faster progress.
- `configs`- Make sure that your arguments are clean and tuned. A `configs.yaml` is a great way to tune your parameters.
- New files- Incase you wish to make a new file for your code, then please do so in the `algos` folder. This will keep the directory consistent.
- `utils`- Any additional utilities such as saving/logging files or plotting results should be implemented as functions in the `utils.py`. This is to help each other while using the code base. I am currently in the process of building this file.