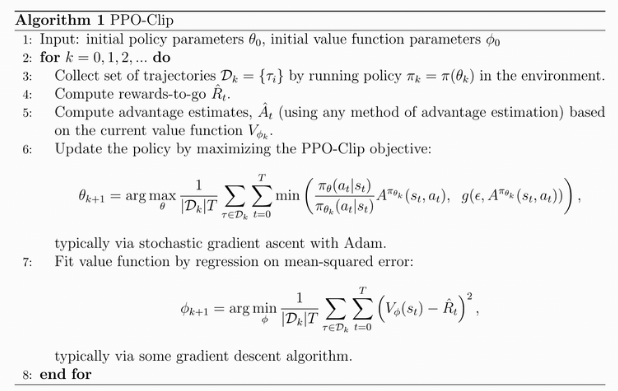
P2 Report

Learning Algorithm:

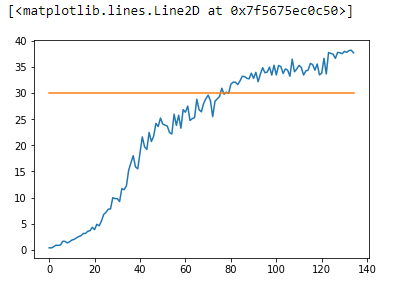
* The learning algorithm is PPO with loss function = clipped surrogate + MSE + entropy. The algorithm includes three neural networks which are actor network, sigma network and critic network. The action network predicts the action, the sigma network predicts the standard deviation of the action and the critic network predicts state value.



* The advantage function can be switched between GAE, TD or Monte Carlo method. In the experiment, I use GAE
* Hyperparameter settings
  + Expected reward discount rate = 0.99
  + GAE discount rate = 0.8
  + Epsilon for ratio clip = 0.2 with decay rate = 0.998
  + Entropy loss weight (beta) = 0.01 with decay rate = 0.99
  + MSE loss weight (mse\_w) = 0.5
  + Buffer tmax = 1000
  + Steps per epoch = 20. For each step, there are 50 mini steps with batch size = 20
* Neural Network Model Architecture
  + Actor network (predict action)
    - Three dense layers. The first two dense layer are followed by relu activation function and the last dense layer is followed by tanh activation function.
    - RELU (Linear (33, 64))
    - RELU (Linear (64, 64))
    - Tanh (Linear (64, 4))
  + Sigma network (predict std of action)
    - Three dense layers. The first two dense layer are followed by relu activation function and the last dense layer is followed by sigmoid activation function.
    - RELU (Linear (33, 64))
    - RELU (Linear (64, 64))
    - Sigmoid (Linear (64, 4))
  + Critic network (predict state value)
    - Three dense layers. The first two dense layer are followed by relu activation function and there is no activation function for the last dense layer.
    - RELU (Linear (33, 64))
    - RELU (Linear (64, 64))
    - Linear (64, 1)

Plot of Rewards:





Ideas of Future Works:

* Can try different algorithm such as DDPG, A3C, or D4PG.
* Can further optimize hyper parameters.
* Can further compare performance among different advantage estimation, GAE, TD and Monte Carlo.
* Try different neural network architecture to see if it speed up training