机器学习纳米学位

《四轴飞行器学会飞行》

2018 年 10 月 23 日

吕金波

目录

1. 定义

1.1 项目概述

四轴飞行器在个人和专业应用领域都变得越来越热门。他易于操控，并广泛应用于各个领域，从最后一公里投递到电影摄影，从杂技表演到搜救，无所不包。

大多数四轴飞行器有 四个提供推力的发动机，虽然有些飞行器有 6 个到 8 个发动机，也叫四轴飞行器。有多个推力点且重力在中心位置改善了稳定性，可以完成多种飞行动作。

但也存在一些问题，此类飞行器控制非常复杂，几乎无法手动操控每个发动机的推力，因此，大部分商业四轴飞行器通过接受一个推力大小和控件简化飞行没这样操作更直观有趣。

1.2 问题陈述

本次项目要解决的问题就是让四轴飞行器能够自动完成一些控制行为，比如简单的起飞，着陆。

在 ROS 与模拟器上训练进行端到端的训练，来完成目标

1.3 评价指标

对于本次项目而言，我们的评价指标是 total_rewards，以及通过模拟器 UI 观察飞行器的运动情况，

```
# TODO: Read and plot episode rewards
landing_path = '/home/bo/workspace/catkin_ws/src/RL-Quadcopter/quad_controller_rl/out/stats_2018-10-21_18-43-04.csv'
plot(landing_path)
```

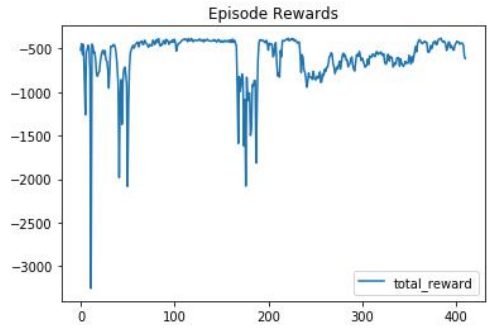


图 1：为 task：landing 训练至收敛的 reward

2. 分析

2.1 **算法完成**

2.1.1DQN

对于连续动作空间上的强化学习任务，我们最先想到的就是 DQN 模型，但是，在本项目中，飞机的状态有 7 个维度，动作有 6 个，也就是说动作空间将无比巨大，在这种情况下，DQN 的训练无法取得很好的效果

而 DeepMind 2015 年提出的 DDPG 可以很好的解决这个问题，所以，本文采取了 DDPG 模型

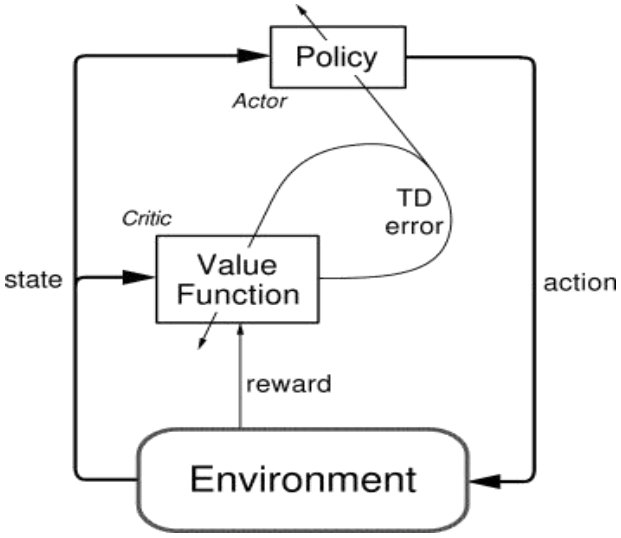
2.1.2 Actor-Critic

Actor Critic 是将 Policy Gradient 和 Q-Learning 等以值为基础的算法组合。

Actor 是一个 Policy Gradient 模型，以 S 为输入，神经网络输出 actions，在通过策略概率模型，选取和是的 action

Critic 是一个 Q-learning 模型，将 Actor 得到的 action 通过与环境交互，的到新的 s’，奖励 r，将 s‘作为神经网络的输入，的到 v’，而原来的 s 通过 v‘由动态规划得到 v。

通过 TD 算法，更新 Actor 网络的参数，实现更新



2.1.3 Buffer-replay

DDPG 主体部分的更新，是更新 Actor-Critic 模型的的参数，这里用到的 Buffer-replay 主要是将没个是 step 的(state, action, reward, next_state) 储存，与用 TD 算法更新参数

3. 具体方法

3.1，状态空间和动作空间分析

机器学习项目的本质是数据，所以，在开始项目之前，先将项目涉及到的数据理顺是重中之重

强化学习的数据量相对其他不值一提，本项目，我们要明确的数据的关键是无人机的状态。

直观的，无人机的状态空间包括（X，Y，Z）的空间坐标系和他的速度，还有方向。

无人机的作业方式是通过几个电机来控制无人机的一系列动作，在上升的 task 中，我们只需要考虑他的位置，在悬浮和着陆中，又需要考虑他状态空间中所有的状态

而动作便是需要网络学习的内容，包括线性力和扭矩

3.2 实现

Actor-Critic 的网络为 4 层全连接层 ,Critic 用了 l2 正则化 ,Buffer-replay 的 Batch 大小为 64 个样本，learning rate 分别为 0.0001,0.0015，gamma=0.99

为了增加鲁棒性，加入了 OU 噪声

在成功实现功能后早期停止

3.2.1 几个 task

对于几个 task，重点在于 reward 的设置:

Task1：想让无人机飞起来是比较容易的，无人机只有受到正向的力且大于重力他才会起飞，而负向力不做移动（当然只是在模拟器中），我们的 reward 设置便可以根据这一点，只要达到目标高度，便基于高额回报。

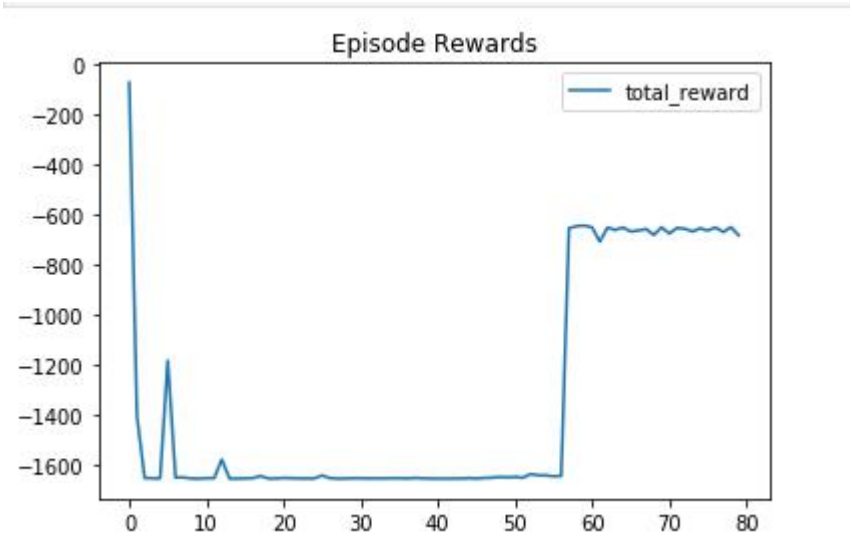
Task2,Task3：这两个 Task 放到一起说，是因为他俩的思路是一致的，与起飞不同的地方是在于，悬浮我们不希望他在控制某一高度乱飞，降落也不希望他以最大速度冲刺下来

对于 Task2 要做的便是限制他的活动范围和速度范围（理想状态当然是保持在目标高度，速度为 0），所以，我们要设置一个容错范围，规定时间内超过这个范围，给予惩罚

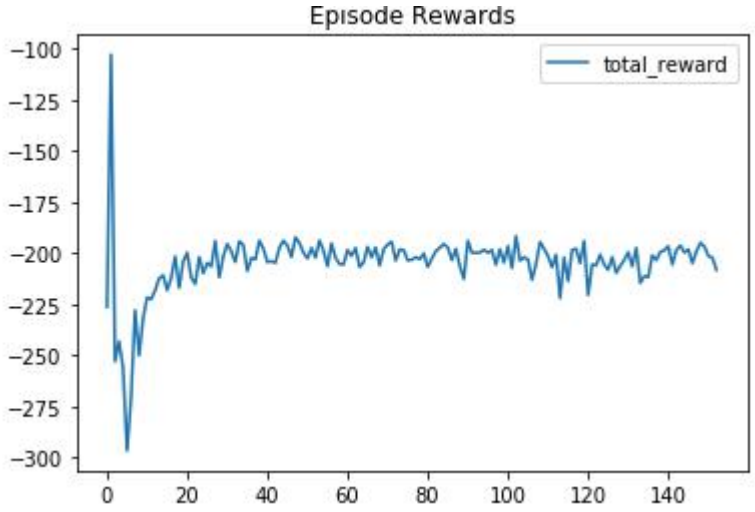
对于 Task3 要做的主要是控制速，不要让他快速落地，所以如果超出容错速度范围，给予惩罚

4. 结论

实验收敛 reward 可视化

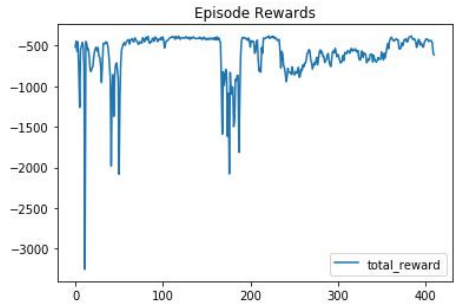


图二：takeoff



图三：Hover

```
# TODO: Read and plot episode rewards
landing_path = '/home/bo/workspace/catkin_ws/src/RL-Quadcopter/quad_controller_rl/out/stats_2018-10-21_18-43-04.csv'
plot(landing_path)
```



图四：Landing

思考：

Combine 模型训练不稳定，需要继续改进。

在实验过程中，我对网络改动很少，只是调了超参，主要精力放在了设计 reward 上了，对网络有过改动但效果不理想，思考是否可以加深网络结果，在全状态空间下学习效果更好。
想做用做一个接管模型，但是动力学基础不足，后续可以考虑做一下。