Openai gym 第三阶段

利用deep-Q-learning实现

一．Q-learning的简单思想

Q-Learning是强化学习算法中value-based的算法，Q即为Q（s,a）就是在某一时刻的 s 状态下(s∈S)，采取 动作a (a∈A)动作能够获得收益的期望，环境会根据agent的动作反馈相应的回报reward r，所以算法的主要思想就是将State与Action构建成一张Q-table来存储Q值，然后根据Q值来选取能够获得最大的收益的动作。

Q-Table a1 a2

s1 q(s1,a1) q(s1,a2)

s2 q(s2,a1) q(s2,a2)

s3 q(s3,a1) q(s3,a2)

1. 设计思路
2. cartpole问题中，我们用obeservation来表示cartpole的状态。四个参数分别为小车在轨道上的位置，杆子与竖直方向的夹角，小车速度，角度变化率。

参数设置：

Observation：

Type: Box(4)

Num Obeservation Min Max

0 Cart Position -2.4 2.4

1 Cart Velocity -Inf Inf

2 Pole Angle -24 deg 24 deg

3 Pole Velocity At Tip -Inf Inf

将其离散化，可得q矩阵

另一个参数action：

Acions：

Type: Discrete(2)

Num Action

1. Push cart to the left

1 Push cart to the left

2.按照以下步骤进行

A.传输当前的状态

B.离散化当前状态，并根据当前状态决定action

C.执行action后，获得新的状态

D.更新q矩阵

1. dqn相对于qlearing的改进：dqn不用q表记录q值，而是用神经网络来预测q值，并通过不断更新神经网络从而学习到最优的行动路径。

dqn中，q值表中表示的是当前已学习到的经验。而根据公式计算出的 q 值是agent通过与环境交互及自身的经验总结得到的一个分数（即：目标 q 值）。最后使用目标 q 值(target\_q)去更新原来旧的 q 值(q)。而目标 q 值与旧的 q值的对应关系，正好是监督学习神经网络中结果值与输出值的对应关系。