1. 回顾 DQN

在 DQN 算法中，我们使用$r+\gamma \max_{a\in A(s')} \hat{q}(s', a, w_T)$来作为 target，使用$\hat{q}(s, a, w)$来作为 predication，计算它们的均方损失来更新网络参数。在 DQN 中，我们使用 target net 来直接计算目标 Q 值，然后再从中选取最大的目标 Q 值，对参数进行更新。

DQN 算法通过贪婪发直接获得目标 Q 值，易导致过估计 Q 值的问题，使模型具有较大的偏差；采用 Double DQN 算法解耦动作的选择和目标 Q 值的计算，以解决过估计 Q 值的问题。

2. Double DQN 原理

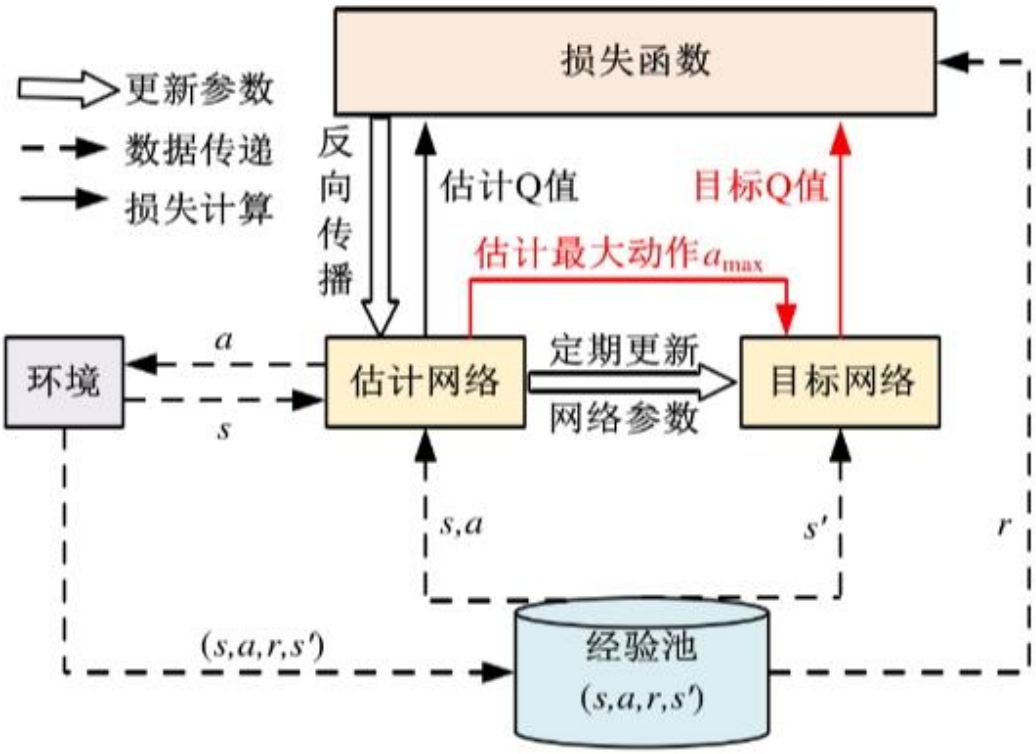
Double DQN 目标 Q 值的计算公式为：

$$y_T = r + \gamma \max_{a'} \hat{q}(s', a', w_T)$$

$$a' = arg\max_{a} \hat{q}(s', a, w)$$

我们首先采用估计网络（同估计预测 q 值的网络），来估计下一个状态 s'的所有 q 值，得到其中 q 值最大的动作 a'，然后使用目标网络计算出下一个状态 s'的所有 q 值，根据刚刚计算得到的 a'而不是通过 max 函数，来选择我们的目标 q 值。然后对目标 q 值和预测 q 值计算损失，对参数进行更新。同样的，我们不对目标网络进行单独更新，而是在每经过一定 step 之后，用估计网络的参数对其进行更新。

算法流程如图所示:


更新参数
数据传递
损失计算
损失函数
反向传播
估计Q值
目标Q值
估计最大动作a_{max}
环境
a
s
估计网络
定期更新
网络参数
目标网络
s,a
s'
r
(s,a,r,s')
经验池
(s,a,r,s')