强化学习：作业三

张三 MG20370001

2020年11月2日

1 作业内容

我们需要在gym Atari环境中实现DQN算法及其变体。本实验的实验环境是Atari Game Pong，Agent需要操控球拍与系统互相击球，未接到球则对方计一分，先取得21分者获胜。实验目标是训练DQN及其变体作为Agent获得游戏胜利，并使获胜时的分差尽可能大。在本次实验中，我分别实现和训练了DQN、Double DQN以及Dueling DQN，并评估了它们在训练过程中的表现和它们在上述游戏中的性能。同时我也实现了Prioritized Replay Buffer，但是完整的算法受限于我的硬件性能无法运行，因此我对论文中的算法进行了一定的简化，并且评估了简化后的算法对DQN训练过程的影响。

2 实现过程

2.1 算法描述

Q-learning 在传统Q-learning算法中，我们使用一张 Q 表来记录环境状态 s 以及该状态对应各个动作 a 的长期回报值 $Q(s,a)$，并使用下式来更新 Q 表：

$$\begin{cases} a' = \arg\max_x Q(s',x) \\ Q(s,a) = Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a)) \end{cases}$$

其中 s' 是在状态 s 下执行动作 a 后的新状态，α 和 γ 分别为学习率和折扣系数。

DQN 在DQN中，我们不使用表型数据结构记录 Q 值，而是用一个深度神经网络来计算不同的状态-动作对应的 Q 值。DQN相较于传统的Q-learning算法能更好地处理状态-动作空间较大的场景。在DQN中，我们需要最小化TD error，既使网络输出 $Q(s,a)$ 逼近于长期回报的估计值 $r +$

$\gamma Q(s', \arg\max_x Q(s', x))$，其中 γ 为折扣系数。我使用均方误差作为损失函数，因此神经网络优化器需要最小化下式：

$$\|r + \gamma Q(s', \arg\max_x Q(s', x)) - Q(s, a)\|_2^2$$

注意到，我们在改变网络参数的时候也会改变优化目标，因此我们需要复制一份原神经网络作为目标网络，从而使优化目标相对稳定。因此改写损失如下：

$$\|r + \gamma Q'(s', \arg\max_x Q'(s', x)) - Q(s, a)\|_2^2$$

其中 Q 为原网络输出，Q' 为目标网络输出。在经过一段时间的训练后，我们需要将原网络参数复制到目标网络上。

Double DQN 该变体是对DQN中训练目标的优化。

Dueling DQN 该变体是对DQN中网络结构的优化。

Prioritized Replay Buffer 优化了Replay Buffer中的采样方式，通过增大TD error较大的样本的采样概率和损失函数权重来提高训练速度。

2.2 代码实现

3 复现方式

3.1 训练复现

复现DQN:在主文件夹下运行 `python atari_ddqn.py --train`.

3.2 测试复现

3.3 参数介绍

复现其他变体:

4 实验效果

4.1 实验图表展示与分析

描述累计奖励和样本训练量之间的关系。
DQN:
其他变体:

5 小结

在这次实验中，我发现...

6 参考文献