强化学习

1 简介

强化学习是机器学习的一个分支，用于搜索智能体与环境交互的最优策略以最大化长期累积奖励。树森师兄近来写了一本强化学习教材，本文结合其中的算法：

1. DQN,
2. SARSA,
3. PER, Double DQN, Dueling Network,
4. REINFORCE,
5. Actor-Critic,
6. Advantage Actor-Critic,
7. TRPO,
8. DPG,
9. TD3,

进行实现层面的介绍。

1.1 马尔可夫决策过程

强化学习问题一般用马尔可夫决策过程(MDP)表示。$M = (S, A, P, r, \rho_0, \gamma)$：

- S 表示状态集合；
- A 表示动作集合；

- P 表示状态转移概率，$P(s' \mid s, a)$ 表示从状态 s，动作 a 转移到 s' 的概率；
- r 表示奖励函数;
- ρ_0 表示初始状态分布；
- γ 表示折扣系数。

强化学习的策略 $\pi(a \mid s)$ 将状态映射到动作空间上的概率分布。智能体与环境交互产生交互轨迹 $\{s_0, a_0, r_0, s_1, a_1, r_1, ...\}$，最优策略最大化累计奖励

$$J(\pi) = \mathbb{E}_{s_0, a_0, r_0, ... \sim \pi} \sum_{i=0}^{\infty} \gamma^i r_i.$$

1.2 值函数与贝尔曼方程

值函数用来衡量当前状态/动作下的长期累积奖励。值函数有两种形式，

- 状态值函数 $V_\pi(s) = \mathbb{E}_{s_0=s, a_0, r_0, s_1, a_1, r_1, ... \sim \pi} \sum_{i=0}^{\infty} \gamma^i r_i$.
- 状态动作值函数 $Q_\pi(s, a) = \mathbb{E}_{s_0=s, a_0=a, r_0, ... \sim \pi} \sum_{i=0}^{\infty} \gamma^i r_i$.

这二者可以互相转换，

$$V_\pi(s) = \sum_a \pi(a \mid s) Q_\pi(s, a),$$
$$Q_\pi(s, a) = r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) V_\pi(s'),$$

二者统称为值函数。

最优策略满足贝尔曼最优方程，

$$V^*(s) = \max_a r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) V^*(s').$$

给定策略 π，对应的值函数满足贝尔曼期望方程，

$$V_\pi(s) = \sum_a \pi(a \mid s)[r(s, a) + \gamma \sum_{s'} P(s' \mid s, a) V_\pi(s')].$$

一个马尔可夫决策过程对应的最优策略可能不是唯一的，但最优值函数是唯一的。给定最优值函数，最优策略可以通过贪心法获得 $\arg\max_a Q(s, a)$。因此，搜索最优策略可以转化成寻找最优值函数。

2 DQN

DQN 是强化学习的一个里程碑算法，大幅提高了强化学习的学习能力 (Mnih et al., 2015)。DQN 使用神经网络参数化值函数 $Q(s, a; \theta)$。最优策略对应的贝尔曼方程可以表示为

$$Q(s, a; \theta^*) = r(s, a) + \gamma \max_{a'} Q(s', a'; \theta^*),$$

对于任意 (s, a, s') 成立。

智能体与环境交互收集很多转移状态 $\{s_i, a_i, r_i, s_i'\}_{i=1}^n$。DQN 最小化以下目标，

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} (Q(s, a; \theta) - (r_i + \gamma \max_{a'} Q(s', a', \theta_{old})))^2.$$

DQN 主要使用了两个技巧来避免训练不收敛，

1. 经验重放（experience replay）。每次训练从中随机抽取 n 个训练样本，降低序贯数据之间的相关性。

2. 延迟目标值函数更新。防止值函数训练过估计（over estimation）。目标值函数和优化值函数使用不同参数的原因是 Q-learning 的想法是动态规划。延迟更新的原因是，因为 DQN 每次随机更新一小部分状态值，不能保证 Q-learning 中的压缩映射，而延迟更新能够从实验上缓解这个问题。请看这里的讨论。

DQN 能够使用过去所有状态转移组的原因是，最优值函数在非最优状态转移下也具有动态规划性质。

CartPole-v0 和 CartPole-v1 上的实验效果如图所示。总体来说，DQN 是一个训练起来不够稳定的算法。

3 SARSA

SARSA 的全称是 State-action-reward-state-sction. 更新方式如下：

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)].$$

其中，$Q(s_{t+1}, a_{t+1})$ 可以继续展开 $Q(s_{t+1}, a_{t+1}) = \sum_{i=1}^{\infty} \gamma^i r_{t+i}$. 这种多步展开的方式广泛用于策略梯度法中。和 DQN 相比，它通过蒙特卡洛采样的方式得到值函数的估计，解除了目标值网络的影响。

Algorithm 1 DQN

初始化值网络和目标值网络。
for $i = 1, 2, ..., N$ **do**
  智能体与环境交互收集转移状态，存入经验缓存中。
  经验缓存中随机抽取批大小为 B 的状态转移集合 $\{s_i, a_i, r_i, s_i'\}$.
  如果 s' 是终止状态，那么目标状态值 $y_i = r_i$；否则，目标状态值 $y_i = r_i + \gamma \max_a Q(s_i', a; \theta_{old})$。
  通过求解优化问题来更新网络参数 $\min_\theta \frac{1}{B} \sum_{i=1}^{B} (Q(s_i, a_i; \theta) - y_i)^2$. 每隔固定步数 C，更新目标网络。
end for

Algorithm 2 SARSA

初始化值网络。
for $i = 1, 2, ..., N$ **do**
  智能体与环境交互收集状态和奖励，存入 M 中。
  if 每收集 C 个 episode 的交互过程 **then**
    将 M 中的状态和奖励取出，按照时间顺序标号 1 到 K，倒序计算对应奖励。$R_{K+1} = 0$.
    for $k = K, K-1, ..., 1$ **do**
      第 k 步的对应奖励为 $R_k = r_k + (1 - done) \cdot \gamma \cdot R_{k+1}$.
    end for
    最小化优化目标 $\frac{1}{K} \sum_{k=1}^{K} (Q(s_k, a_k; \theta) - R_k)^2$.
  end if
end for

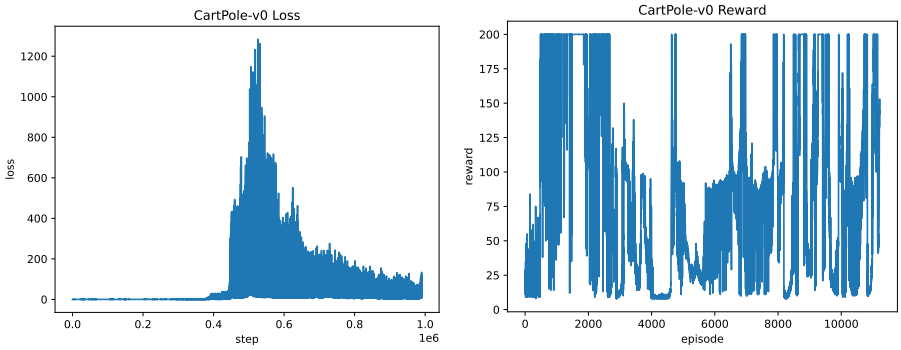

图 1: DQN 在 CartPole-v0 上效果。

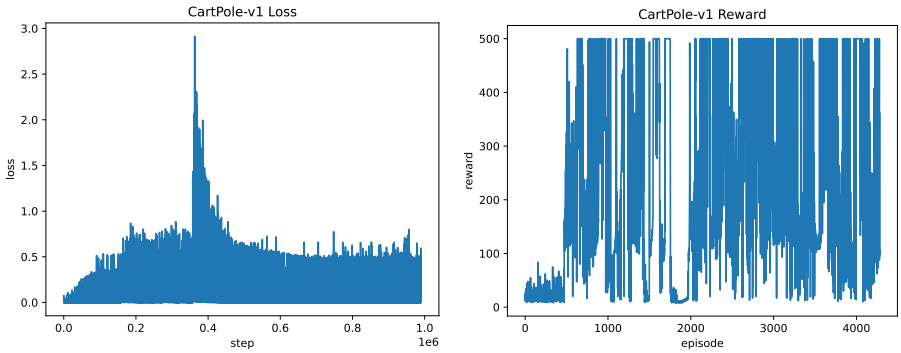

图 2: DQN 在 CartPole-v1 上效果。

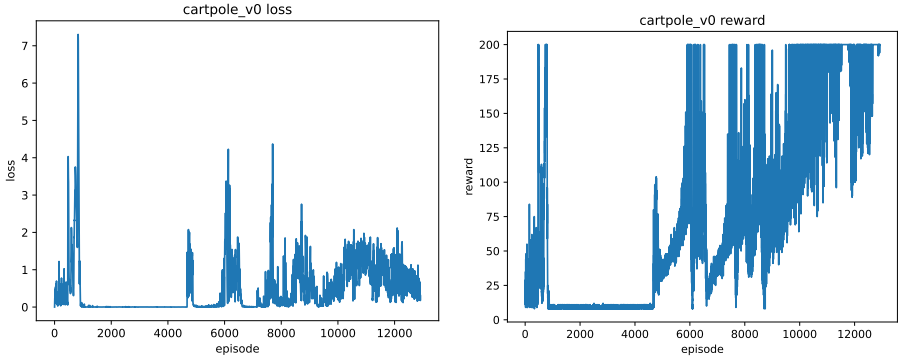

图 3: SARSA 在 CartPole-v0 上效果。

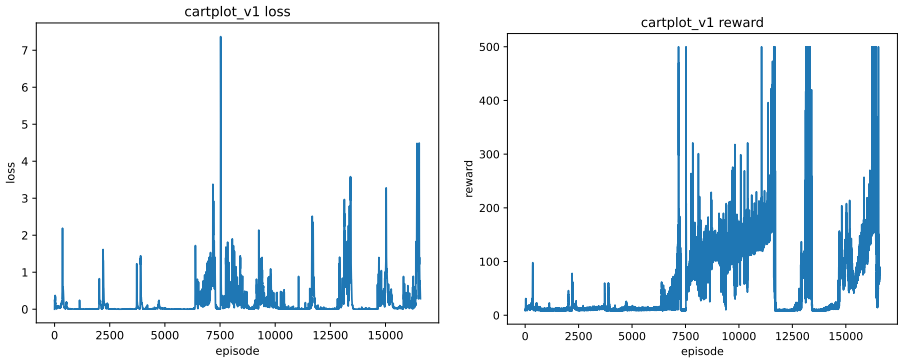

图 4: SARSA 在 CartPole-v1 上的效果。

4 PER, Double DQN, Dueling Network

DQN 训练不够稳定，数据利用率不高。有三个小技巧来提升 DQN 训练：

- Prioritized Experience Replay，即优先经验回放；
- Double DQN;
- Dueling Network.

4.1 优先经验回放

强化学习方法从经验中学习，优先经验回放方法通过调整不同样本的重要程度，来改进优化目标的损失函数

参考文献

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.