8 - DQN 算法进阶

Double DQN 算法

Double DQN 算法是谷歌 DeepMind 于 2015 年 12 月提出的一篇论文，主要贡献是通过引入两个网络用于解决 Q 值过估计（ overestimate ）的问题。

先回顾一下 DQN 算法的更新公式，如式 (8.1) 所示。

$$Q_\theta(s_t,a_t) \leftarrow Q_\theta(s_t,a_t)+\alpha[r_t+\gamma \max_a Q_{\hat{\theta}}(s_{t+1},a_{t+1})-Q_\theta(s_t,a_t)]$$

其中 $y_t=r_t+\gamma \max_a Q_{\hat{\theta}}(s_{t+1},a_{t+1})$ 是估计值，这里会使用目标网络各个动作对应的最大 Q 值来当做估计值，这样一来会导致过估计的问题存在。为了解决这个问题 Double DQN 提出利用当前最大 Q 值对应的动作没然后再将其带入目标网络中去计算 Q 值。

$$a_\theta^{\max}(s_{t+1})=\arg\max_a Q_\theta(s_{t+1},a)$$

$$y_t=r_t+\gamma \max_a Q_{\hat{\theta}}(s_{t+1},a_\theta^{\max}(s_{t+1}))$$

Dueling DQN 算法

在 Double DQN 算法中我们是通过改进目标 Q 值的计算来优化算法的，而在 Dueling DQN 算法中则是通过优化神经网络的结构来优化算法的。

DQN的网络结构：

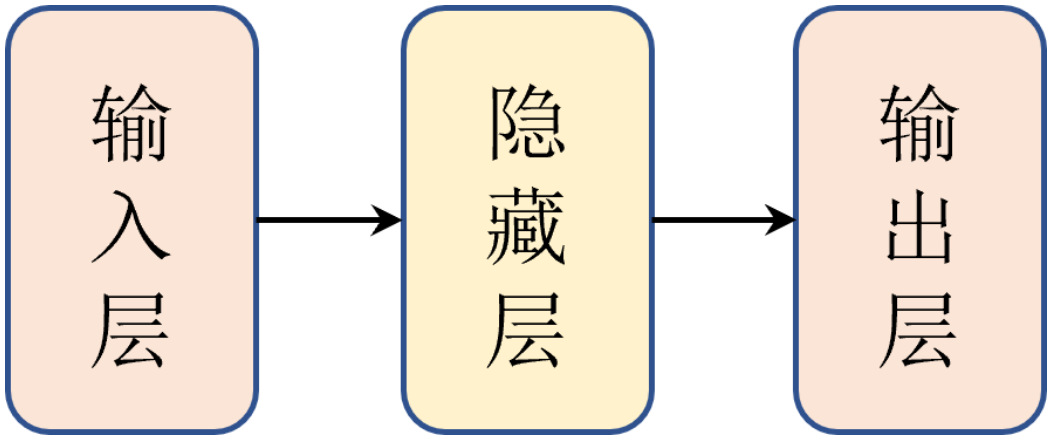

图 8-1 DQN 网络结构

Deuling DQN 的网络结构：

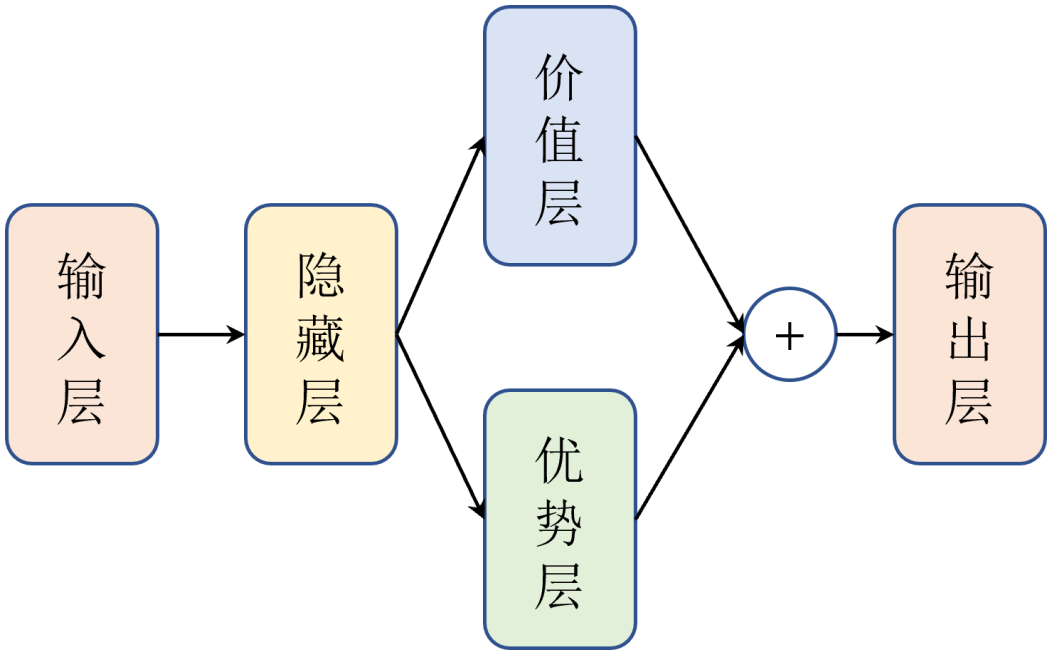

图 8-2 Dueling DQN 网络结构

在DQN算法中我们使用 $Q_\theta(s,a)$ 表示一个 Q 网络，在这里我们使用 $A_{\theta,\alpha}(s,a)$ 来表示优势层，而相对应的价值层为 $V_{\theta,\beta}(s)$。因此 Deuling DQN的网络结构能够表示为

$$Q_{\theta,\alpha,\beta}(s,a)=A_{\theta,\alpha}(s,a)+V_{\theta,\beta}(s)$$

然后对优势层做一个中心化处理，即减掉均值：

$$Q_{\theta,\alpha,\beta}(s,a)=(A_{\theta,\alpha}(s,a)-\frac{1}{A}\sum_{a\in A}A_{\theta,\alpha}(s,a))+V_{\theta,\beta}(s)$$

总的来讲，Dueling DQN 算法在某些情况下相对于 DQN 是有好处的，因为它分开评估每个状态的价值以及某个状态下采取某个动作的 Q 值。

Noisy DQN 算法

Noisy DQN 算法其实是在 DQN 算法基础上在神经网络中引入了噪声层来提高网络性能的，即将随机性应用到神经网络中的参数或者说权重，增加了 Q 网络对于状态和动作空间的探索能力，从而提高收敛速度和稳定性。

PER DQN 算法

PER DQN 算法优化了原有在DQN算法中的经验回放。PER DQN算法中的经验回放加入了优先级的设定，具体实现就是由TD误差来得到各个节点的优先级，我们通常使用SumTree这样的二叉树结构进行实现。

这样会出现两个问题：第一，如果某批样本的TD误差过低时会导致网络不更新，可能会导致到错过高信息量的样本。第二，这个优先级排序容易导致只有反复那几个样本被挑选为更新样本。

为了解决这两个问题，我们引入随机优先级采样的技巧，即每次更新是，我们不直接采样TD误差最大的样本，卫视定义一个采样的概率

$$P(i)=\frac{p_i^a}{\sum_k p_k^a}$$

其中，p_i 是样本 i 的优先级， α 是超参数。为了避免最低优先级样本的采样概率不为0，因此我们可以在优先级加上一个常数 ϵ：

$$p_i = |\delta_i| + \epsilon$$

其中，$|\delta_i|$ 是样本 i 的TD误差。

除了随机优先级采样之外，我们还引入了重要性采样。这个采样方法是一种用于估计某一分布性质的方法，它的基本思想是，我们可以通过与待估计分布不同的另一个分布中采样，然后通过采样样本的权重来估计待估计分布的性质，其表达式为：

$$\begin{aligned}\mathbb{E}_{x\sim p(x)}[f(x)] &= \int f(x)dx \\ &= \int f(x)\frac{p(x)}{q(x)}q(x)dx \\ &= \int f(x)\frac{p(x)}{q(x)}\frac{q(x)}{p(x)}p(x)dx \\ &= \mathbb{E}_{x\sim p(x)}[\frac{p(x)}{q(x)}f(x)]\end{aligned}$$

其中 $p(x)$ 是待估计分布，$q(x)$ 是采样分布，$f(x)$ 是待估计分布的性质。在前面我们讲到，每次计算的 TD 误差是对应之前的网络，而不是当前待更新的网络。也就是说，我们已经从之前的网络中采样了一批样本，也就是 $q(x)$ 已知，然后只要找到之前网络分布与当前网络分布之前的权重 $\frac{p(s)}{q(x)}$，就可以利用重要性采样来估计出当前网络的性质。由此我们可以定义权重为：

$$w_i = (\frac{1}{N}\frac{1}{P(i)})^{\beta}$$

其中，N 是经验回放中的样本数量，$P(i)$ 是样本 i 的采样概率。同时，为了避免出现权重过大或过小的情况，我们可以对权重进行归一化处理。

$$w_i = \frac{(N * P(i))^{-\beta}}{\max_j(w_j)}$$

$$\beta = \min(1, \beta + \beta_{\text{step}})$$

练习题

DQN 算法为什么会产生 Q 值的过估计问题?

DQN只用最大化的 Q 值导致了在收到噪声和训练过程中的不稳定。

同样是提高探索，Noisy DQN 和 ϵ-greedy 策略 有什么区别?

Noisy DQN 使用了噪声而 ϵ-greedy 使用了固定的 ϵ。