深度Q网络（DQN）

主讲老师：枫老师

· 2022 ·

主要内容

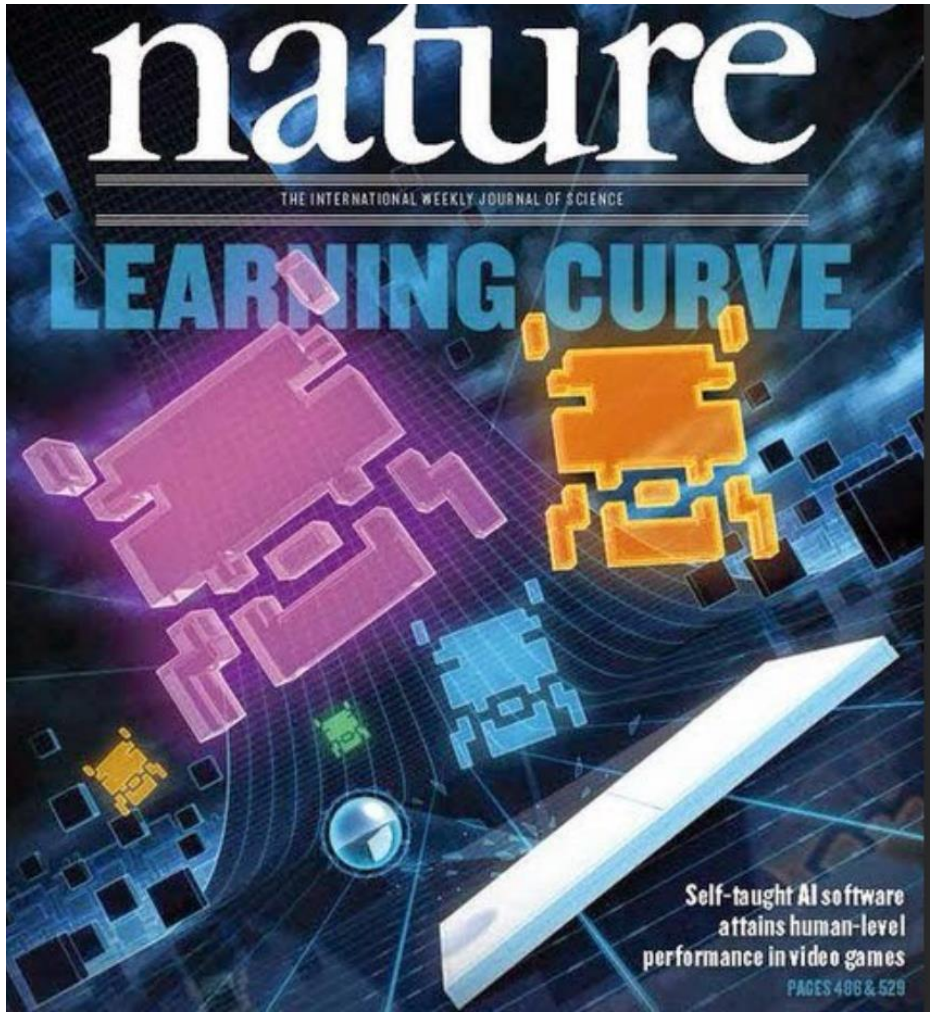
1. DQN原理

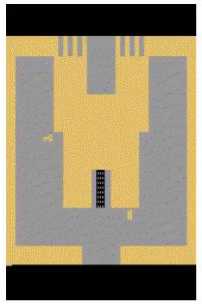
- 神经网络拟合Q函数
- 经验重放池
- 带延迟的目标网络

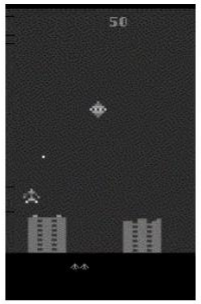
2. DQN实战

- DQN智能体的设计
- 经验重放池的实现、学习过程实现
- 编程实战：构建DQN智能体玩月球车着陆游戏

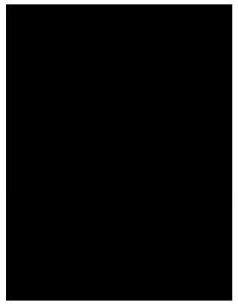
DQN发表于《Nature》杂志

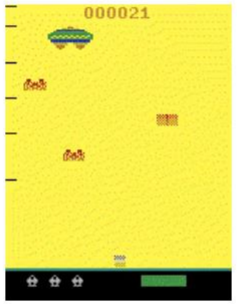


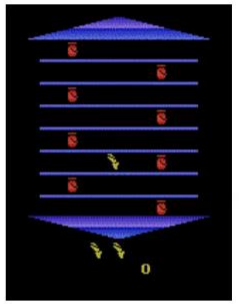

Adventure

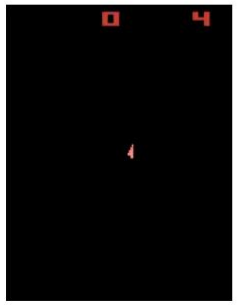

Air Raid


Alien


Amidar


Assault


Asterix


Asteroids


Atlantis


Bank Heist

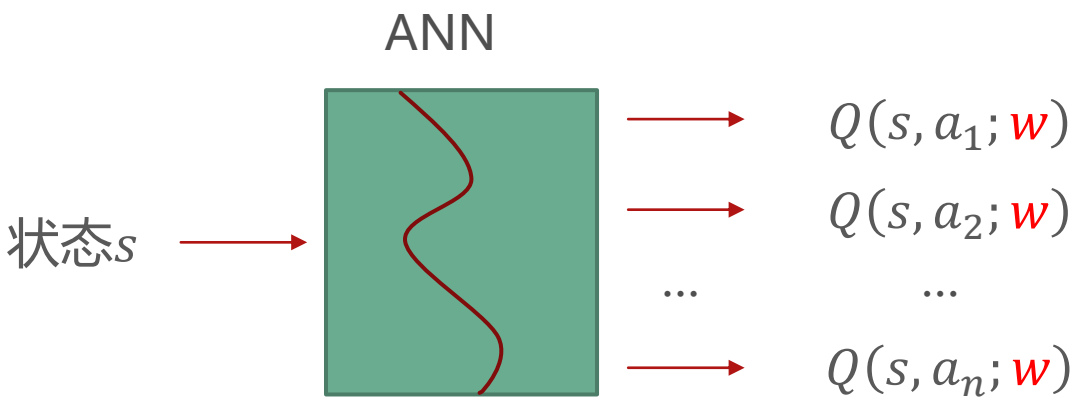
《Human-level control through deep reinforcement learning》

神经网络拟合Q函数

- 从表格型Q值表到神经网络拟合Q函数（NFQ）

动作a

状态s			
	$Q(s_1, a_1)$	$Q(s_1, a_1)$	$Q(s_1, a_1)$
	$Q(s_2, a)$	$Q(s_2, a_2)$	$Q(s_2, a_2)$
	$Q(s_3, a)$	$Q(s_3, a_3)$	$Q(s_3, a_3)$



表格型Q-learning

$$\pi(a|s) = \underset{a'}{\operatorname{argmax}}\, Q(s, a')$$

神经网络拟合Q函数

$$\pi(a|s) = \underset{a'}{\operatorname{argmax}}\, Q(s, a'; w)$$

如何应对超大规模状态空间/连续状态空间?

内存?

时间?

Riedmiller, M. (2005). Neural fitted Q iteration - First experiences with a data efficient neural Reinforcement Learning method. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3720 LNAI*, 317–328.

神经网络拟合Q函数

- NFQ的学习过程

回顾Q-learning算法的更新过程：

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

现在Q函数从一张表格变成了一个神经网络(参数为w)。如何需学习？

神经网络的参数w决定了Q函数的输出特性。因此学习过程就是调整参数w。

如果神经网络表征的Q函数对未来的期望收益足够准确，那么时间差分误差理论上为0，即

$$\delta = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; w) - Q(S_t, A_t; w) = 0$$

因此构建优化目标函数

$$\delta^2 = \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; w) - Q(S_t, A_t; w)\right]^2$$

使用梯度下降算法优化神经网络参数w

神经网络拟合Q函数

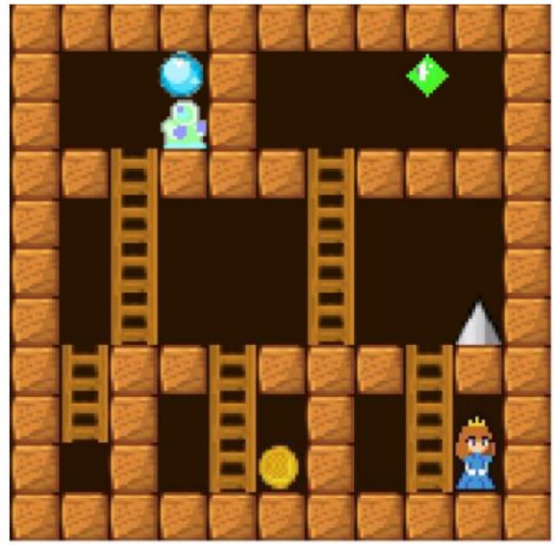
- NFQ存在的问题——训练不稳定

使用梯度下降更新神经网络参数：

$$\delta^2 = \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; w) - Q(S_t, A_t; w\right]^2$$

$$\nabla_w \delta^2 = 2\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; w) - Q(S_t, A_t; w)\right] \nabla_w Q(S_t, A_t; w)$$

$$w = w - \alpha \nabla_w \delta^2$$



图片来自Context-aware policy reuse. AAMAS2019

这里面涉及 $< S_t, A_t, R_{t+1}, S_{t+1}, >$，即智能体与环境交互的一个样本，样本集合用$\mathcal{D}$表示。

（1）训练过程一般会分批多次从$\mathcal{D}$中取样本训练神经网络，样本之间存在较大的相关性。

（2）训练过程TD target和Q函数同时更新导致训练不稳定。

DQN的改进

- DQN针对NFQ训练的不稳定进行了改进

主要使用了两方面的改进：

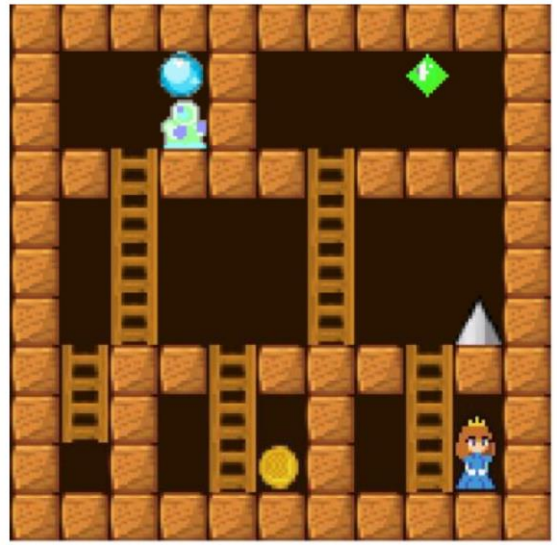
（1）从$\mathcal{D}$中随机选择样本进行训练，降低样本之间的相关性。

（2）固定TD target神经网络的参数，并定期和Q函数神经网络进行同步。

$$\nabla_w \delta^2 = 2\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; w^-) - Q(S_t, A_t; w)\right] \nabla_w Q(S_t, A_t; w)$$

$$\delta^2 = \left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a; w^-) - Q(S_t, A_t; w\right]^2$$

$$w = w - \alpha \nabla_w \delta^2$$

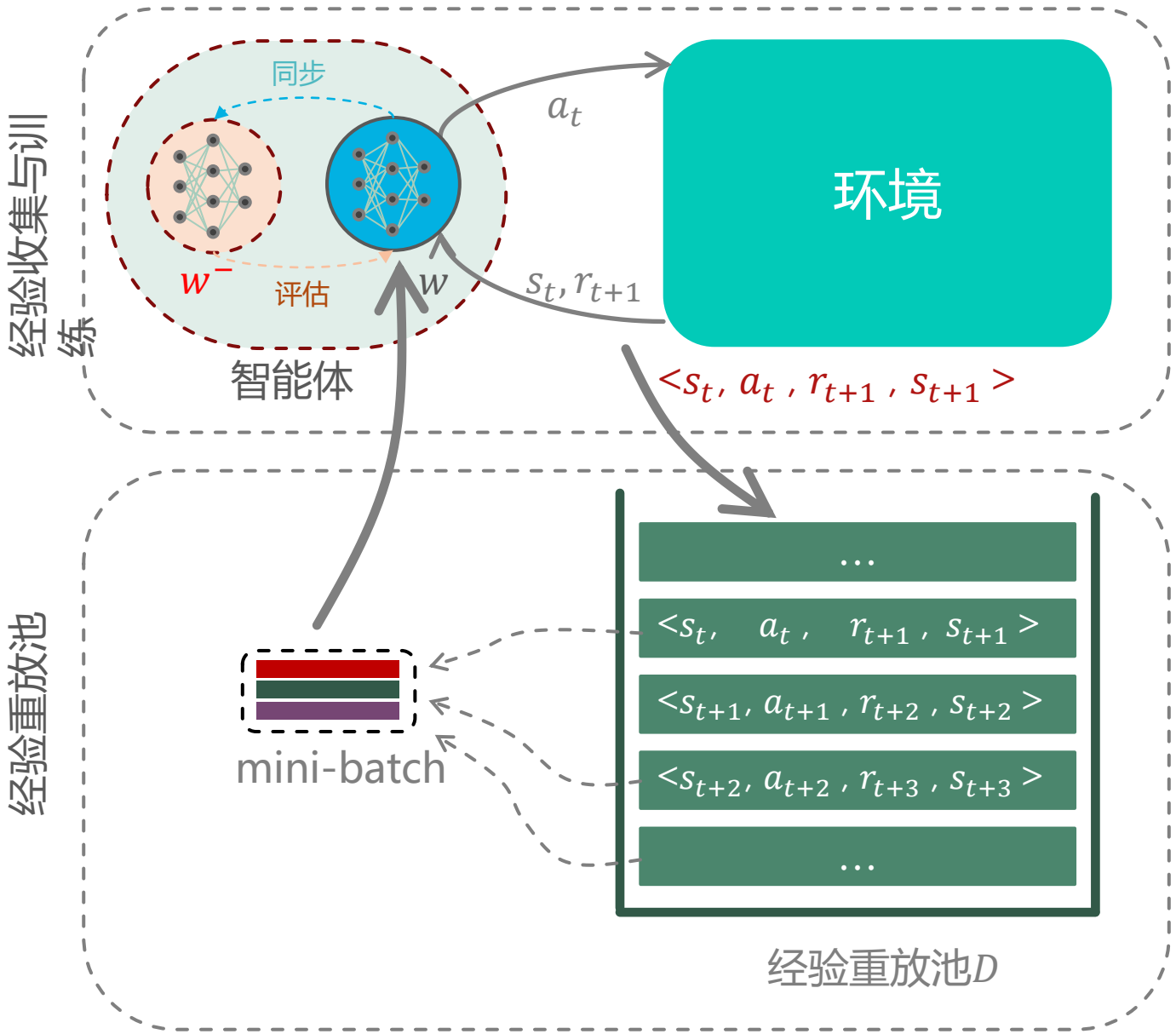


图片来自Context-aware policy reuse. AAMAS2019

主要内容

Agent类
- Q网络
- 延迟Q网络
- act()函数
- learn()函数

QNetwork类
- layers
- forward()函数



ReplayBuffer类
- memory
- buffer_szie
- sample()函数

主要内容

1. DQN原理

- 神经网络拟合Q函数
- 经验重放池
- 带延迟的目标网络

2. DQN实战

- DQN智能体的设计
- 经验重放池的实现、学习过程实现
- 编程实战：构建DQN智能体玩月球车着陆游戏