强化学习“探索与利用”方法研究

洪星星

2020年11月16日

目录

背景知识

MDP$\langle S, A, P, R, \gamma \rangle$

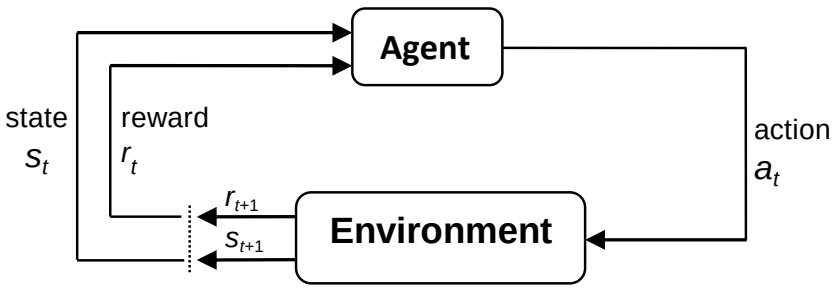


图 1: 智能体与环境交互示意

- S和A分别为状态和动作集合
- $P(s, a, s')$是转移概率函数
- $R(s, a)$是即时收益函数
- 依据策略$\pi(s, a)$与环境交互产生轨迹
  $\boldsymbol{h} = [s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_t, a_t, r_{t+1}, \ldots]$
- 累计收益 $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$
  $(0 \leq \gamma \leq 1)$
- 状态价值函数 $V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$
- 状态-动作价值函数
  $Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a]$

策略评估

- 策略评估 评估策略π下的期望累积收益$V^{\pi}(s)$及$Q^{\pi}(s,a)$
- 贝尔曼期望等式

$$V^{\pi}(s) = \Sigma_a \pi(a|s) \Sigma_{s'} P\left(s,a,s'\right)\left(R\left(s,a,s'\right)+\gamma V^{\pi}\left(s'\right)\right)$$

$$Q^{\pi}(s,a) = \Sigma_{s'} P\left(s,a,s'\right)\left(R\left(s,a,s'\right)+\gamma \Sigma_{a'} \pi\left(a'|s'\right) Q^{\pi}\left(s',a'\right)\right)$$

策略评估

Theorem

基于$V^{\pi}(s)=\Sigma_a\pi(a|s)\Sigma_{s'}P(s,a,s')\left(R(s,a,s')+\gamma V^{\pi}(s')\right)$，定义

- $R^{\pi}_{ave}(s)=\Sigma_a\pi(a|s)\Sigma_{s'}P(s,a,s')\,R(s,a,s')$
- $T^{\pi}(s\rightarrow s')=\Sigma_a\pi(a|s)P(s,a,s')$

定义算子$\mathcal{T}$，可证明$\mathcal{T}$为*Banach*空间中的压缩算子

$$\mathcal{T}V\equiv R^{\pi}_{ave}+\gamma T^{\pi}V$$

策略评估

Theorem

贝尔曼期望等式递推展开，得$V^{\pi}(\vec{S})$与$Q^{\pi}(s,a)$表达式

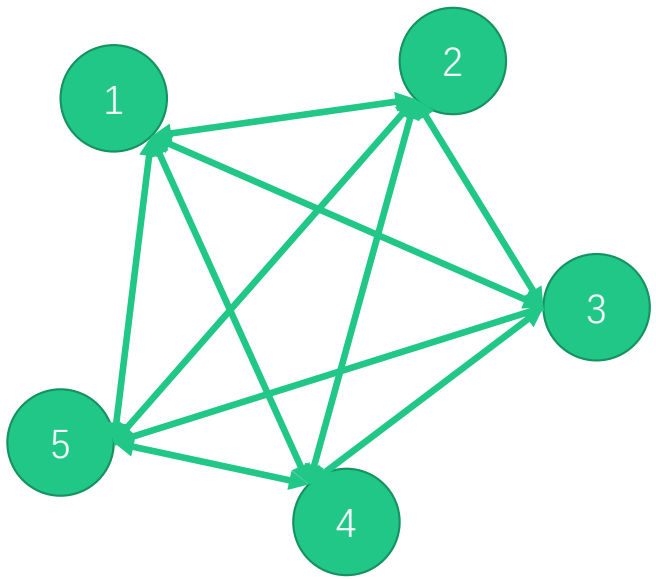
$$
\begin{aligned}
V^{\pi}(\vec{S}) &= R^{\pi}_{ave}(\vec{S}) + (\gamma T^{\pi})^{1} \cdot R^{\pi}_{ave}(\vec{S}) + (\gamma T^{\pi})^{2} \cdot R^{\pi}_{ave}(\vec{S}) + \dots \\
&= \sum_{k=0}^{\infty} (\gamma T^{\pi})^{k} \cdot R^{\pi}_{ave}(\vec{S}) \overset{①}{=} \sum_{k=0}^{\infty} (\gamma T^{\pi})^{(k)} \cdot R^{\pi}_{ave}(\vec{S}) \\
&\overset{②}{=} (\boldsymbol{I} - \gamma T^{\pi})^{-1} \cdot R^{\pi}_{ave}(\vec{S}) \\
Q^{\pi}(s,a) &= R(s,a) + \gamma P(s,a,\vec{S})(\boldsymbol{I} - \gamma T^{\pi})^{-1} R^{\pi}_{ave}(\vec{S})
\end{aligned}
$$

思考题

- ①和②为什么成立?
- 求$V^{\pi}(\vec{S})$对π矩阵的梯度

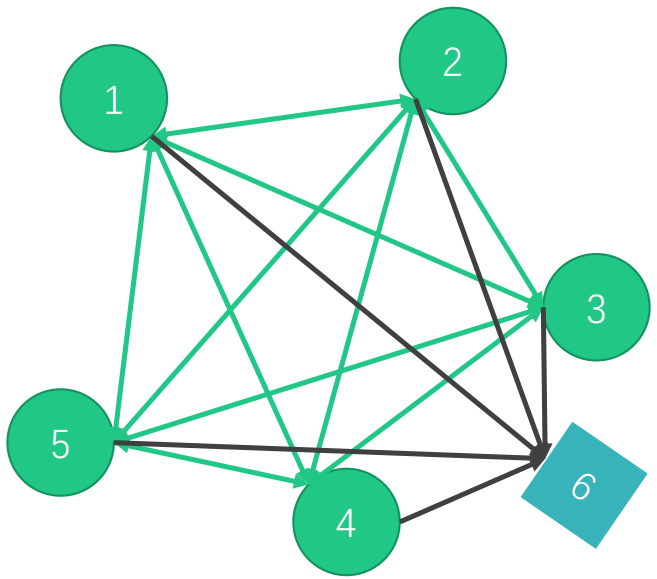
$$\frac{\partial}{\partial \pi} V^{\pi}(\vec{S})$$

策略评估



MDP

{1,2,3,4,5} 之间有正的跳转概率;
{1,2,3,4,5}为常返态集合



MDP'

{1,2,3,4,5}之间的正概率按γ衰减;
{1,2,3,4,5}变成非常返态集合;
新引入的终止态{6}为常返态集合

图 2: 引入$\gamma(0 < \gamma < 1)$可视作把MDP转换为MDP'的示例

策略优化

- 策略优化 找最优策略π^*以最大化期望累积收益$V^{\pi}(s)$

$$\pi^* = \arg\max_{\pi} \mathbb{E}[G_t]$$

- 贝尔曼最优化等式

$$V^*(s) = \max_{\pi} V^{\pi}(s) = \max_{a} \Sigma_{s'} P(s,a,s')\left(R(s,a,s') + \gamma V^*(s')\right)$$

$$Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a) = \Sigma_{s'} P(s,a,s')\left(R(s,a,s') + \gamma \max_{a'} Q^*(s',a')\right)$$

$$V^*(s) = \max_{a} Q^*(s,a)$$ 最优策略必为确定性策略

- 类似定义贝尔曼最优化算子具有压缩性以证明存在唯一不动点

策略优化

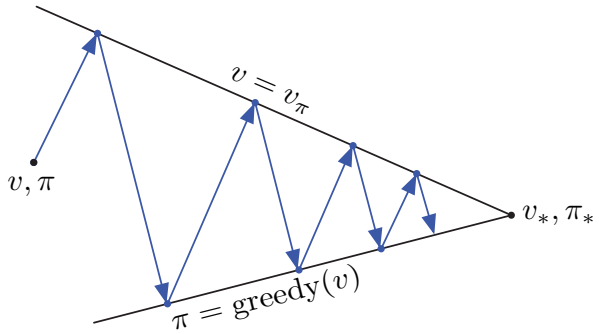


图 3: 广义策略提升框架GPI中策略评估与优化交替(Sutton and Barto, 2018)

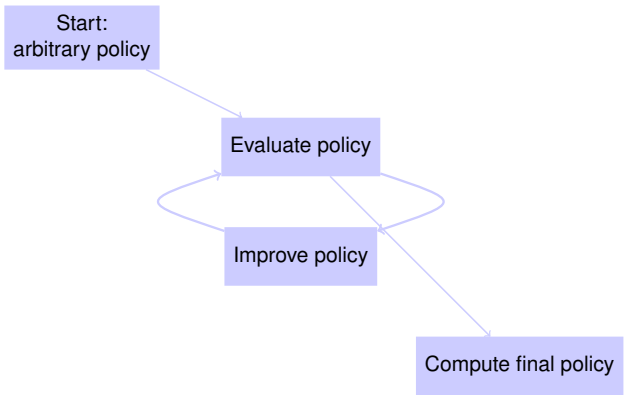


图 4: 策略迭代

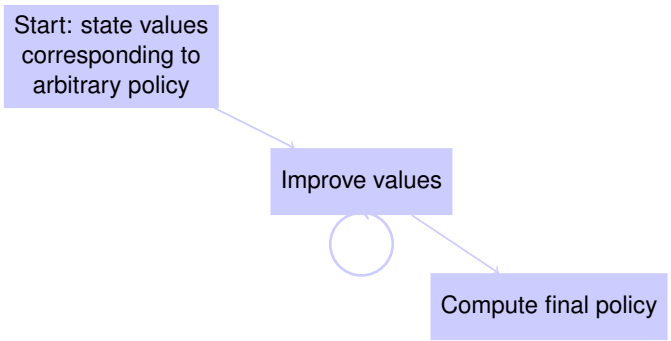


图 5: 值迭代

策略优化

策略从π提升到π'常采用greedy, $\epsilon-$greedy和softmax几种方式

- greedy

$$\pi'(a|s) = \begin{cases} 1 & a = \arg\max_a Q^\pi(s,a) \\ 0 & \text{otherwise} \end{cases}$$

- $\epsilon-$greedy

$$\pi'(a|s) = \begin{cases} \frac{\epsilon}{|A|} + 1 - \epsilon, & a = \arg\max_a Q^\pi(s,a) \\ \frac{\epsilon}{|A|}, & \text{otherwise} \end{cases}$$

- softmax

$$\pi'(a|s) = \frac{e^{\beta Q^\pi(s,a)}}{\sum_{a \in A} e^{\beta Q^\pi(s,a)}} \quad 0 \leq \beta;$$

蒙特卡洛方法

蒙特卡洛智能体与环境交互产生若干条状态，动作与收益序列$\boldsymbol{h} = [s_0, a_0, r_1, s_1, a_1, r_2, \ldots, s_t, a_t, r_{t+1}, \ldots]$。在计算值函数时，蒙特卡洛方法利用经验池中$\boldsymbol{h}$带衰减累计收益的均值估计$V^{\pi}(s) = \mathbb{E}[G_t | s_t = s]$期望值，而能否准确估计值函数，则依赖算法高效地探索和利用经验。N表示从s出发的轨迹条数，$G_j(s)$表示第j条轨迹的带衰减累计收益值，可采用递增技巧计算N条经验轨迹的均值。

$$
\begin{aligned}
V_N(s) &= \frac{1}{N}\sum_{j=1}^{N} G_j(s) \\
&= \frac{1}{N}\left(G_N(s) + \sum_{j=1}^{N-1} G_j(s)\right) \\
&= \frac{1}{N}\left(G_N(s) + (N-1)V_{N-1}(s)\right) \\
&= V_{N-1}(s) + \frac{1}{N}\left(G_N(s) - V_{N-1}(s)\right)
\end{aligned}
$$

更一般的蒙特卡洛更新形式如下，其中α为学习率。

$$V(s) \leftarrow V(s) + \alpha\left(G_t - V(s)\right)$$

也即 $V(s) \leftarrow \alpha G_t + (1-\alpha)V(s)$

时序差分方法

采用bootstrap及在线学习思想，它估计下一状态-动作值函数，用α控制值函数更新的学习率，更新公式如下。其中$r_{t+1}+\gamma Q(s_{t+1},a_{t+1})$被称为TD目标（TD target），它与$Q(s_t,a_t)$作差称作TD误差（TD error）。

$$Q(s_t,a_t)\leftarrow Q(s_t,a_t)+\alpha\left[\overbrace{\underbrace{r_{t+1}+\gamma Q(s_{t+1},a_{t+1})}_{\text{TD target}}-Q(s_t,a_t)}^{\text{TD error}}\right]$$

状态s_{t+1}依据不同的动作采样与Q值计算方法确定$Q(s_{t+1},a_{t+1})$，可导出Q学习与SARSA等变种算法。

时序差分方法

- Q学习

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a_{t+1} \in A} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

- SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1} | a_{t+1} \sim \epsilon - greedy) - Q(s_t, a_t) \right]$$

- Expected SARSA

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \mathbb{E}_\pi \left[Q(s_{t+1}, a_{t+1}) | s_{t+1} \right] - Q(s_t, a_t) \right]$$

实际计算将依据ϵ-greedy策略采样出M个动作a_{t+1}，计算M个Q值的均值$\bar{Q}$替代期望$\mathbb{E}$运算，故Expected SARSA亦可写成以下形式。

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \bar{Q}_{a_{t+1} \sim \epsilon - greedy} (s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

深度强化学习

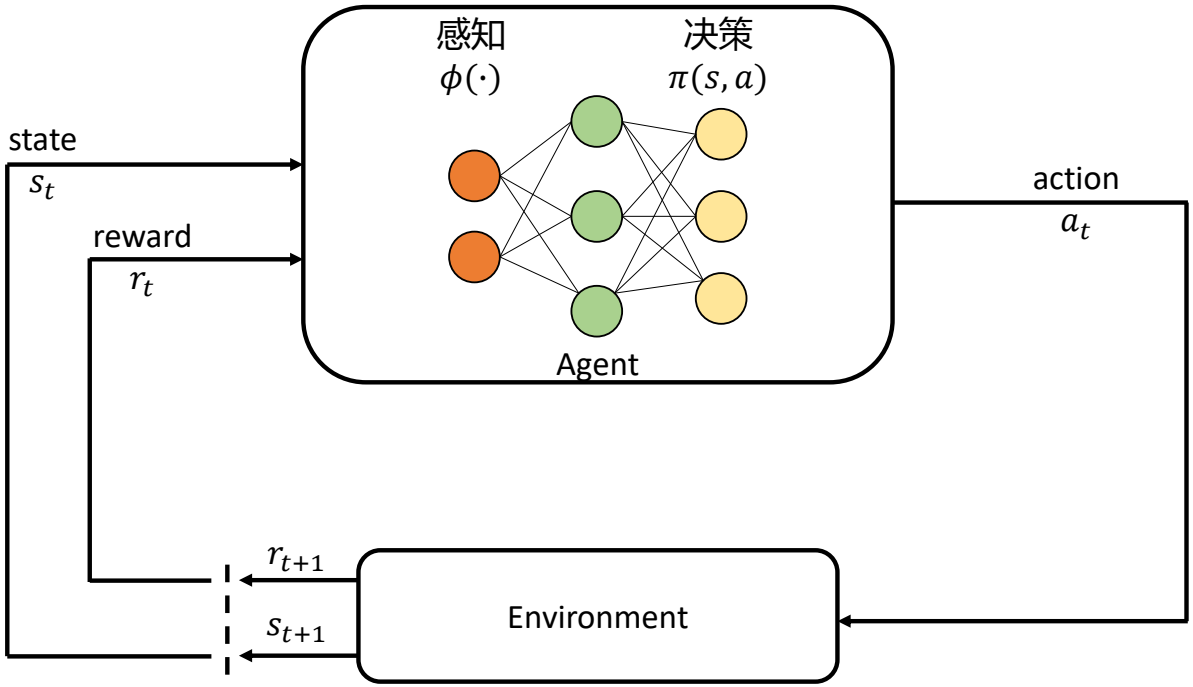
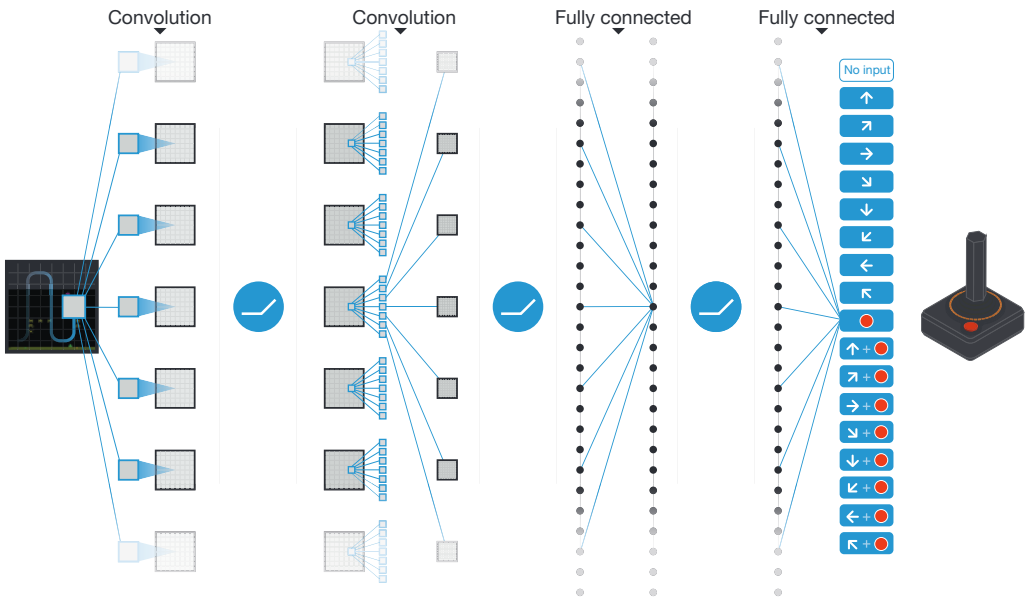


图 6: 深度强化学习智能体与环境交互示意

- 强化学习算法分类
  - 基于值函数拟合、基于策略函数拟合
  - 基于模型、免模型

强化学习算法

DQN



- 深度卷积网络
- 经验回放
- 目标网络

图 7: DQN架构(Mnih et al., 2015)

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim \mathrm{U}(D)}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right) - Q\left(s, a; \theta_i\right)\right)^2\right]$$

$$\nabla_{\theta_i} L(\theta_i) = \mathbb{E}_{s,a,r,s'}\left[\left(r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right) - Q\left(s, a; \theta_i\right)\right) \nabla_{\theta_i} Q\left(s, a; \theta_i\right)\right]$$

DQN变种

表 1: DQN及其变种算法

算法	特点
DQN (Mnih et al., 2015)	CNN网络，卷积层感知，全连接层输出动作概率； 经验回放池机制； 目标网络以稳定训练； $Y^{DQN} \equiv r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right)$
Double DQN (van Hasselt et al., 2016)	将选动作和评估值函数解耦，用两个网络 $Y^{DoubleDQN} \equiv r + \gamma Q\left(s', \arg\max_{a' \in A} Q(s', a', \theta_i); \theta_i^-\right)$
优先经验回放 (Schaul et al., 2016)	将相对更重要的记忆以更高频率被回放
Dueling DQN (Wang et al., 2016)	定义优势函数$A^{\pi}(s,a)$，将$Q^{\pi}(s,a)$拆解为$V^{\pi}(s)$与$A^{\pi}(s,a)$两部分
Rainbow (Hessel et al., 2018)	结合上述所有特点的方案，还包括多步、随机参数等技巧

Double DQN

$$Y^{DQN} \equiv r + \gamma \max_{a'} Q\left(s', a'; \theta_i^-\right)$$

Double DQN和DQN一样，也有两个Q网络。DQN的第二个Q网络起目标网络的作用，DDQN的第二个网络发挥了将选动作与评估该状态-动作值函数解耦的作用。当选定动作a后，评估值函数仍采用$Q\left(s', a'; \theta_i^-\right)$，但选动作则依据$\arg\max_{a' \in A} Q\left(s', a', \theta_i\right)$，DDQN的TD目标为$Y^{DoubleDQN}$。$\theta_i$与$\theta_i^-$表示两个$Q$网络的参数。

$$Y^{DoubleDQN} \equiv r + \gamma Q\left(s', \arg\max_{a' \in A} Q(s', a', \theta_i); \theta_i^-\right)$$

PER. DQN

随机优先抽样方法介于纯贪心抽样和均匀随机抽样之间，设转移样本i处的TD偏差为δ_i，设定样本i的采样概率$P(i)$为

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

p_i为转移样本i的优先级，转移样本形如$\langle s_i, a_i, r_{i+1}, s_{i+1}, a_{i+1} \rangle$。指数$\alpha$指示优先级对采样概率的重要度，当$\alpha = 0$时退化成均匀随机采样。有两种方法定义$p_i$，一种为$p_i = |\delta_i| + \epsilon$，$\epsilon$为小的正数以确保TD偏差为0的样本也有一定概率被抽中；第二种为$p_i = \frac{1}{\text{rank}(i)}$，其中$\text{rank}(i)$表示将所有样本按$|\delta_i|$排序存储后样本$i$的序号。用$\delta_i$确定$P(i)$，因为TD误差越大，样本越能让智能体“惊奇”，该样本包含的“信息量”越大。

Dueling DQN

将全连接层拆分为二：一支计算状态值函数$\hat{V}(s)$，另一支计算关于动作优势值$\hat{A}(s,.)$，最后再汇合为$\hat{Q}(s,.)$状态-动作值函数。

$$Q^{\pi}(s,a) = V^{\pi}(s) + A^{\pi}(s,a)$$

为导出上式中$Q^{\pi}(s,a)$、$V^{\pi}(s)$和$A^{\pi}(s,a)$之间的关系，推导过程如下。

$$V^{\pi}(s) = \sum_{a} \pi(s,a;\theta) Q^{\pi}(s,a)$$

$$\sum_{a} \pi(s,a;\theta) V^{\pi}(s) = \sum_{a} \pi(s,a;\theta) Q^{\pi}(s,a)$$

$$0 = \sum_{a} \pi(s,a;\theta) \left[Q^{\pi}(s,a) - V^{\pi}(s) \right]$$

Dueling DQN

将项$Q^{\pi}(s,a) - V^{\pi}(s)$定义为优势函数$A^{\pi}(s,a)$，它度量出相较于平均动作，某动作$a$具有的相对优势。

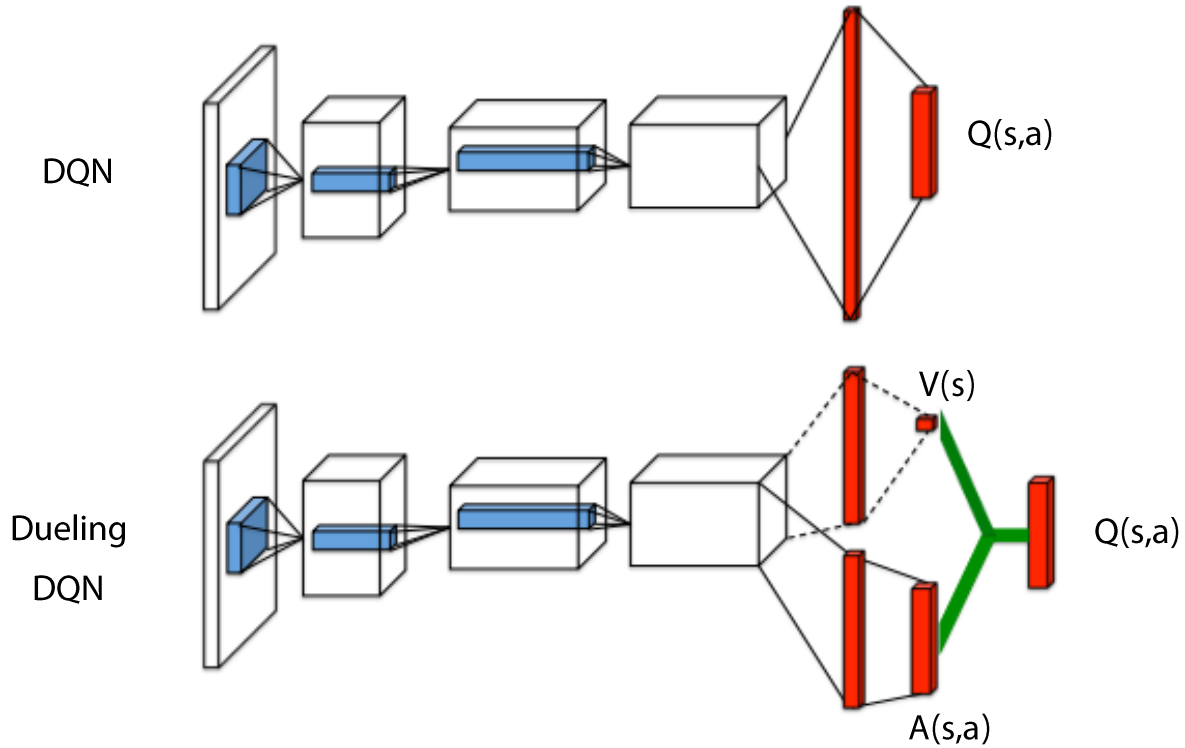


图 8: Dueling DQN架构

随机策略梯度定理

$$
\begin{aligned}
\nabla V^{\pi}(s) &= \nabla \left[\sum_{a} \pi(s,a;\theta) Q^{\pi}(s,a) \right], \quad \text{对所有} s \in S \\
&= \sum_{a} \left[\nabla \pi(s,a;\theta) Q^{\pi}(s,a) + \pi(s,a;\theta) \nabla Q^{\pi}(s,a) \right] \\
&= \sum_{a} \left[\nabla \pi(s,a;\theta) Q^{\pi}(s,a) + \pi(s,a;\theta) \nabla \sum_{s'} P\left(s,a,s'\right) \left[r(s,a,s') + \gamma V^{\pi}\left(s'\right) \right] \right] \\
&= \sum_{a} \left[\nabla \pi(s,a;\theta) Q^{\pi}(s,a) + \gamma \pi(s,a;\theta) \sum_{s'} P\left(s,a,s'\right) \underbrace{\nabla V^{\pi}\left(s'\right)}_{\text{递推展开}} \right] \\
&= \sum_{a} \Bigg[\nabla \pi(s,a;\theta) Q^{\pi}(s,a) + \gamma \pi(s,a;\theta) \sum_{s'} P\left(s,a,s'\right) \times \\
&\qquad \underbrace{\sum_{a'} \left[\nabla \pi(s',a';\theta) Q^{\pi}(s',a') + \gamma \pi(s',a';\theta) \sum_{s''} P(s',a',s'') \nabla V^{\pi}(s'') \right]}_{} \Bigg] \\
&= \ldots \\
&= \sum_{x} \sum_{k=0} (\gamma T^{\pi})^{(k)}_{s \to x} \sum_{a} \left[\nabla \pi(x,a;\theta) Q^{\pi}(x,a) \right] \\
&= \sum_{x} \sum_{k=0} (\gamma T^{\pi})^{(k)}_{s \to x} \sum_{a} \left[\frac{\nabla \pi(x,a;\theta)}{\pi(x,a;\theta)} \pi(x,a;\theta) Q^{\pi}(x,a) \right] \\
&= \sum_{x} \sum_{k=0} (\gamma T^{\pi})^{(k)}_{s \to x} \mathbb{E}_{\pi} \left[Q^{\pi}(x,a) \nabla \log \pi(x,a;\theta) \right]
\end{aligned}
$$

思考题

- 蒙特卡洛方法估计梯度

$$\begin{aligned}\nabla J(\theta) &= \sum_s p_1(s) V^\pi(s) \\ &= \sum_s p_1(s) \sum_x \sum_{k=0} (\gamma T^\pi)^{(k)}_{s\to x} \mathbb{E}_\pi \left[Q^\pi(x,a) \nabla \log \pi(x,a;\theta) \right] \\ &\cdots \\ &\approx \frac{1}{m} \sum_{i=1}^{m} \sum_{k=0}^{H_i} \left[\gamma^k Q(x,a) \nabla \log \pi(x,a;\theta) | x_0 \xrightarrow{(k)} x \right]\end{aligned}$$

随机策略梯度算法

表 2: REINFORCE等随机策略梯度算法

算法	特点
REINFORCE (Sutton et al., 2000)	基于随机梯度定理，梯度为 $\sum_x \sum_{k=0} (\gamma T^\pi)^{(k)}_{s\to x} \mathbb{E}_\pi [Q^\pi(x,a) \nabla \log \pi(x,a;\theta)]$ 用蒙特卡洛方法估计梯度 $\frac{1}{m} \sum_{i=1}^{m} \sum_{k=0}^{H_i} \left[\gamma^k Q(x,a) \nabla \log \pi(x,a;\theta) \mid x_0 \xrightarrow{(k)} x \right]$
引入基线的REINFORCE	引入常数基数b有利于减小方差 且对策略梯度保持为无偏估计
Actor-Critic	策略函数拟合之外增加对状态-动作价值函数拟合
A2C (Mnih et al., 2016)	把Critic网络原始的累计收益替换成优势函数
A3C (Mnih et al., 2016)	多线程异步运行 全局有一套Actor神经网络参数 各环境中也存有一副本

确定性策略梯度定理

Theorem

$$
\begin{aligned}
\nabla_\theta V^{\mu_\theta}(s) &= \nabla_\theta Q^{\mu_\theta}\left(s, \mu_\theta(s)\right) \\
&= \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p\left(s \to s', t, \mu_\theta\right) \nabla_\theta \mu_\theta\left(s'\right) \nabla_a Q^{\mu_\theta}\left(s', a\right)\Big|_{a=\mu_\theta(s')} \mathrm{d}s' \\
\nabla_\theta J\left(\mu_\theta\right) &= \nabla_\theta \int_{\mathcal{S}} p_1(s) V^{\mu_\theta}(s) \mathrm{d}s \\
&= \int_{\mathcal{S}} p_1(s) \nabla_\theta V^{\mu_\theta}(s) \mathrm{d}s \\
&= \int_{\mathcal{S}} \int_{\mathcal{S}} \sum_{t=0}^{\infty} \gamma^t p_1(s) p\left(s \to s', t, \mu_\theta\right) \nabla_\theta \mu_\theta\left(s'\right) \nabla_a Q^{\mu_\theta}\left(s', a\right)\Big|_{a=\mu_\theta(s')} \mathrm{d}s' \mathrm{d}s \\
&= \int_{\mathcal{S}} \rho^{\mu_\theta}(s) \nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu_\theta}(s, a)\Big|_{a=\mu_\theta(s)} \mathrm{d}s
\end{aligned}
$$

- 结论引自(*Silver et al., 2014*)，与随机策略梯度推导有异曲同工之妙

确定性策略梯度算法

表 3: DDPG等确定性策略梯度算法

算法	特点
DPG (Silver et al., 2014)	基于确定性策略梯度定理 状态映射到动作值而非动作概率分布$a = \mu_\theta(s)$ $\int_{\mathcal{S}} \rho^{\mu_\theta}(s) \nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu_\theta}(s,a)\big\|_{a=\mu_\theta(s)} \mathrm{d}s$ $\theta^{k+1} = \theta^k + \alpha \mathbb{E}_{s\sim\rho^\mu}\left[\nabla_\theta \mu_\theta(s) \nabla_a Q^{\mu^k}(s,a)\big\|_{a=\mu_\theta(s)}\right]$
异策略DPG (Silver et al., 2014)	利用行为策略的样本辅助训练 解决确定性策略探索性不足的问题
DDPG (Lillicrap et al., 2015)	借用DQN两个机制：经验回放和独立目标网络 目标网络的参数采用soft更新模式

基于信赖域思想的策略梯度算法

表 4: TRPO与PPO算法

算法	特点
TRPO (Schulman et al., 2015)	基于信赖域优化思想 用变化前后的策略的KL散度确定参数更新邻域 控制策略迭代的更新幅度 再求解梯度更新方向与步长 $J(\theta) = \mathbb{E}_{s\sim\rho^{\pi_{\theta_{\text{old}}}}, a\sim\pi_{\theta_{\text{old}}}}\left[\frac{\pi_\theta(a\|s)}{\pi_{\theta_{\text{old}}}(a\|s)}\hat{A}_{\theta_{\text{old}}}(s,a)\right]$ $\mathbb{E}_{s\sim\rho^{\pi_{\theta_{\text{old}}}}}[D_{\text{KL}}(\pi_{\theta_{\text{old}}}(.\|s)\|\pi_\theta(.\|s)] \leq \delta$
PPO (Schulman et al., 2017)	用近端策略优化思想简化目标函数的约束求解 $J^{\text{CLIP}}(\theta) = \mathbb{E}[\min(r(\theta)\hat{A}_{\theta_{\text{old}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{\theta_{\text{old}}}(s,a))]$

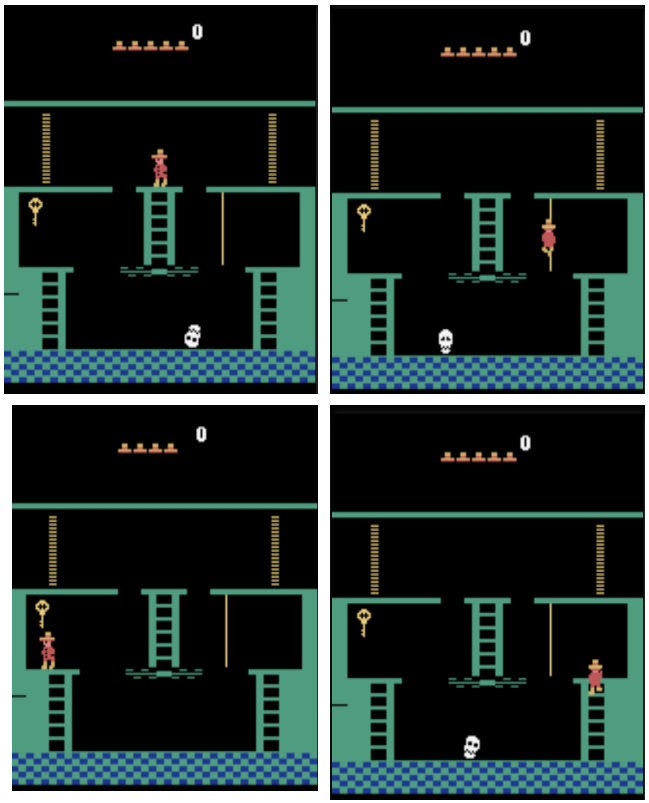
研究动机

策略优化

强化学习算法策略提升(优化)非常重要

- γ
  - γ可被视作正则项(Amit et al., 2020)
  - γ性质探究(Pitis, 2019)
- 策略提升算子(softmax)
  - softmax算子不具有非扩张性(Littman and Szepesvari, 1996)
  - 替代策略提升算子Mellowmax(Asadi and Littman, 2017)
  - Dynamic Boltzmann Softmax(Pan et al., 2019)
- 策略梯度定理与优化方法
  - 矩阵表示、求导与优化方法
  - TRPO(Schulman et al., 2015)和ACKTR(Wu et al., 2017)等算法

硬探索(Hard Exploration Problem)



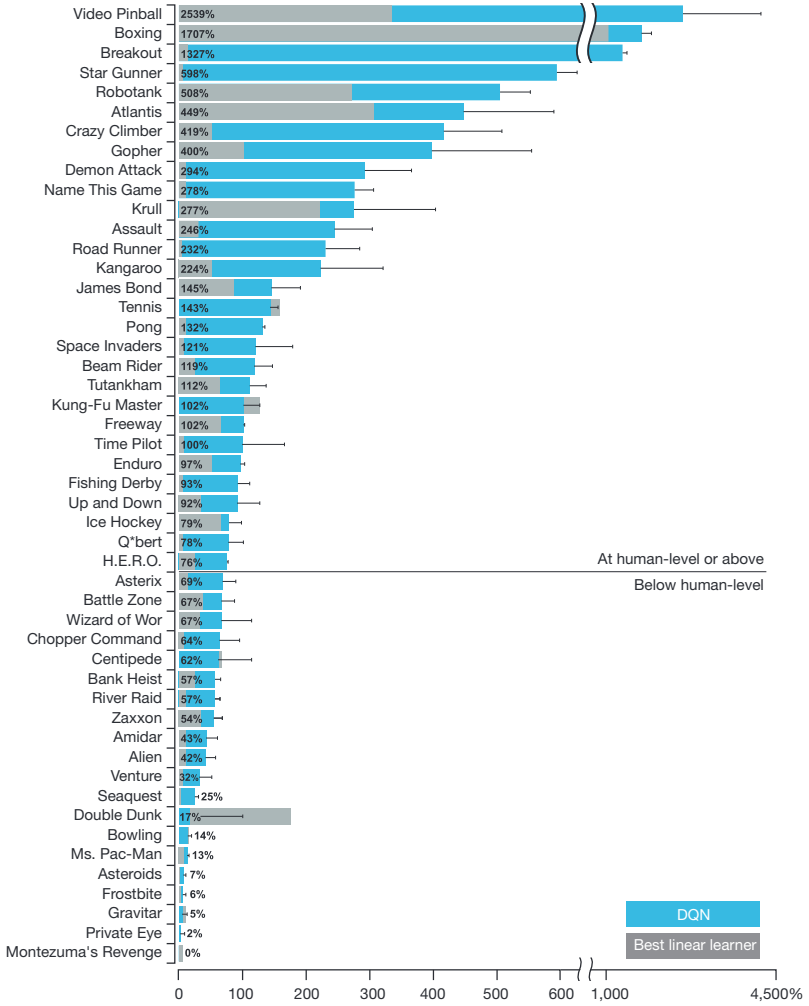


图 9: DQN(Mnih et al., 2015)

样本效率(Sample Inefficiency)

- 1000万游戏帧训练DQN

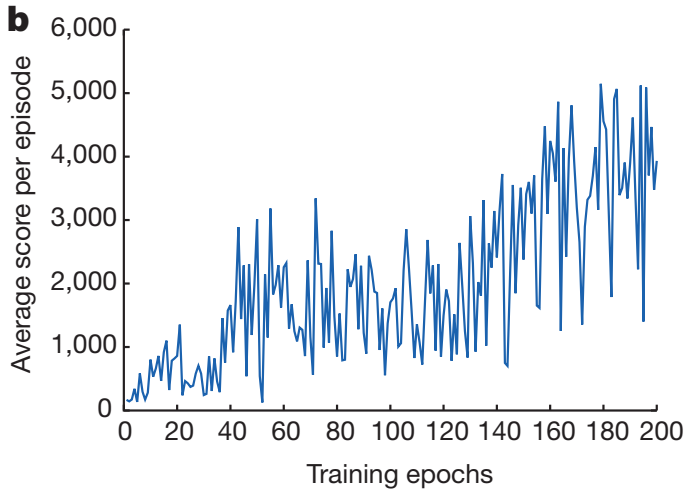


图 10: Atari Seaquest(Mnih et al., 2015)

- 490万局围棋对局

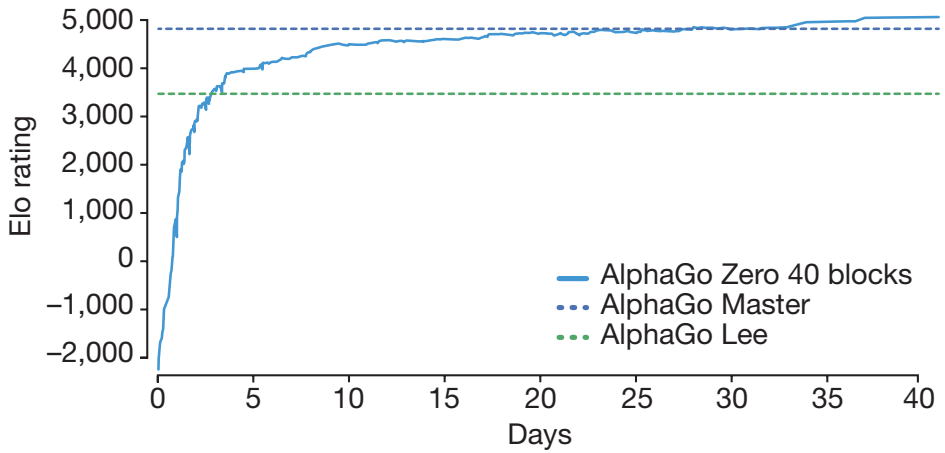
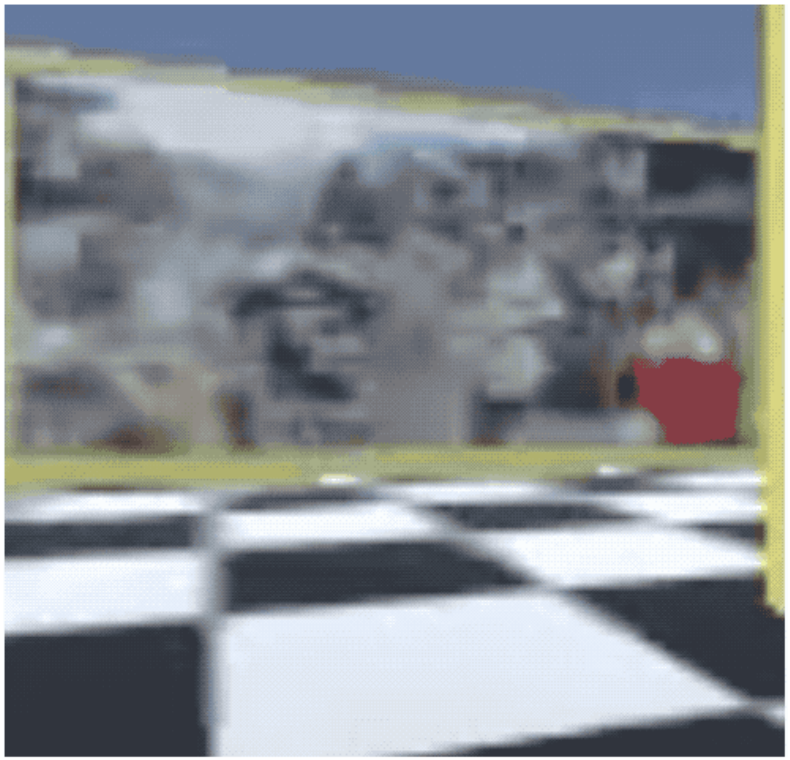
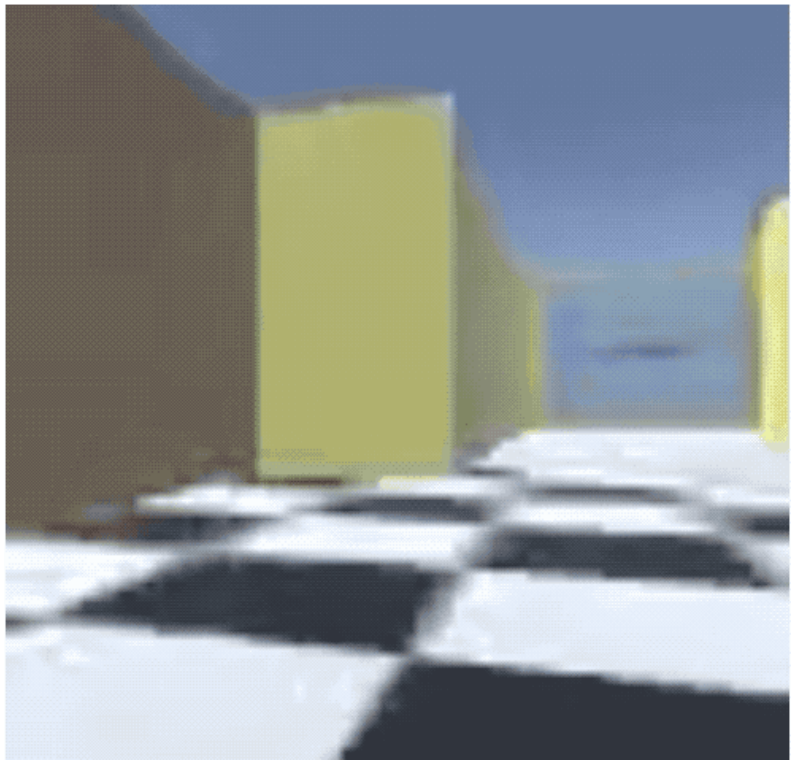


图 11: AlphaGo(Silver et al., 2017)

环境噪声(Noisy-TV Problem)



Agent in a maze with a noisy TV



Agent in a maze without a noisy TV

图 12: 迷宫墙面上的噪声电视(Noisy-TV Problem)(Burda et al., 2019)

相关数学基础

相关数学基础

- 马尔科夫链(He, 2008)(Kang, 2015)
  - K-C方程、常返性、可约性、周期性、遍历性与遍历定理等
- 线性代数(Fang et al., 2013)
  - 范数、赋范线性空间、谱半径、特征值上界定理、转移概率矩阵特征值、矩阵幂级数收敛充要条件、压缩映射原理等
- 概率论与统计学(Casella and Berger, 2007)(Downey, 2013)(Orloff and Bloom)(Duchi)
  - 离散型与连续型先验的贝叶斯定理、Beta分布、马尔科夫不等式、车比雪夫不等式、霍夫丁引理、切诺夫霍夫丁界等
- 信息论(Q. et al., 2006)
  - 熵、相对熵、互信息、信息增益、条件熵、联合熵、凸函数、Jensen不等式、max函数凸性、熵的凹性等
- 数值优化(Gao, 2014)
  - 信赖域优化方法

“探索与利用”概述

探索与利用

- 探索问题(Levine, 2018)
  - 智能体如何发现高收益策略，往往最优策略由时序上一系列复杂行动构成，且很可能单独考察每个行动并无正收益?
  - 智能体该如何决定：是该尝试新行动来发现更高收益，抑或用已知能够获取高收益的行动来继续应对?
- 两种权衡(Zhang, 2019)
  - 探索与利用困境(Exploration and Exploitation Dilemma)
  - 为将来利用而探索(Exploration for Future Exploitation)
- 探索策略(Zhang, 2019)
  - 随机探索
  - 系统性探索

多臂老虎机

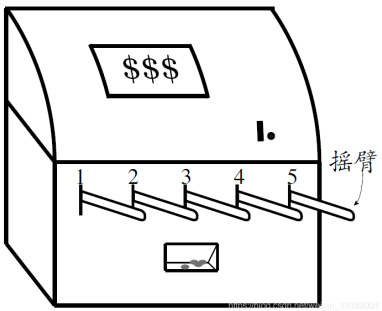


图 13: MAB

Algorithm 1: Multi-Armed Bandit (MAB) Learning

for $t = 1; \ldots; T$: **do**

Algorithm selects an arm $i_t \in [n]$;

Simultaneously, environment selects a reward vector $\mathbf{x}^t = (x_1^t, \ldots, x_n^t) \in [0, 1]^n$;

Algorithm observes reward $x_{i_t}^t$;

end

图 14: MAB Learning(Agarwal, 2013)

基于乐观探索的方法

- 平均遗憾值函数 T次摇动最优老虎机臂的期望收益和−实际动作序列对应的期望收益和

$$\mathrm{Reg}(T) = T\mu^* - \sum_{t=1}^{T} \mathbb{E}\left[\mu_{i_t}\right]$$

- UCB1算法(Auer et al., 2002)
  - $i_t \leftarrow \mathrm{argmax}_{i\in[n]}\left(\widehat{\mu}_i^{t-1} + \sqrt{\frac{\alpha \ln t}{2N_i^{t-1}}}\right)$
  - $\mathrm{Reg}(T)$上界为$O(\log T)$

基于汤普森采样的方法

- 使用后验概率而非$\epsilon-$greedy做更有针对性的探索(Chapelle and Li, 2011)
- Beta分布为0-1分布的共轭先验，在先验分布为Beta分布而似然函数为0-1分布时，后验概率分布仍是Beta分布

$$\theta_i \sim B(S_i + \alpha, F_i + \beta)$$

$$S_{\hat{i}} = S_{\hat{i}} + 1, \text{如果} r_{i \sim \text{Beta}} = 1$$

$$F_{\hat{i}} = F_{\hat{i}} + 1, \text{如果} r_{i \sim \text{Beta}} = 0$$

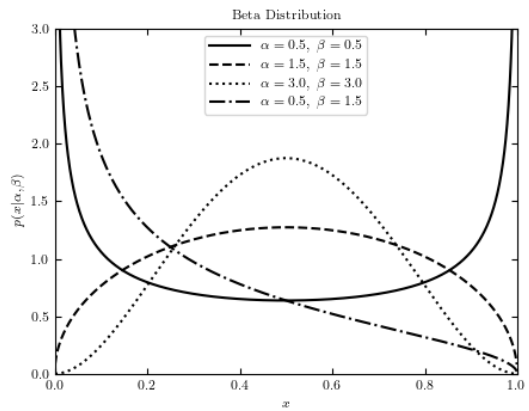


图 15: Beta分布(Vanderplas et al., 2012)

基于信息增益的方法

- (Russo and Roy, 2014)从信息增益角度提出探索策略IDS(Information Directed Sampling)

$$\begin{array}{ccccc} \mathscr{F}_t = & (A_1, & Y_{1,A_1}, \ldots, & A_{t-1}, & Y_{t-1,A_{t-1}}) \\ & \downarrow & \cdots & \downarrow & \\ & R_{1,A_1} & \cdots & R_{t-1,A_{t-1}} & \end{array}$$

- 关于A^*的后验概率分布 $\alpha_t(a) = \mathbb{P}(A^* = a|\mathscr{F}_t)$
- 信息增益 $g_t(a) = \mathbb{E}[H(\alpha_t) - H(\alpha_{t+1})|\mathscr{F}_t, A_t = a]$
- 在t时选择a的即时遗憾期望 $\Delta_t(a) := \mathbb{E}\left[R_{t,A^*} - R_{t,a}|\mathscr{F}_t\right]$

基于信息增益的方法

- π策略下的平均信息增益

$$\pi \in \mathscr{D}(\mathscr{A}),\ \ g_t(\pi) := \Sigma_{a\in\mathscr{A}}\pi(a)g_t(a)$$

- π策略下的平均遗憾期望

$$\Delta_t(\pi) = \sum_{a\in\mathscr{A}} \pi(a)\Delta_t(a)$$

- 探索策略IDS(Information Directed Sampling)

$$\pi_t^{\mathrm{IDS}} \in \arg\min_{\pi\in\mathcal{D}(\mathscr{A})} \left\{ \Psi_t(\pi) := \frac{\Delta_t(\pi)^2}{g_t(\pi)} \right\}$$

- 避免那些几乎获取不到新信息的选择
- 避免那些明显次优的选择

探索方法的设计原则

探索方法设计的惯用原则(Levine, 2018)(Weng, 2020)

1. 增加随机扰动
2. 引入熵正则项以增大探索性
3. 对未知抱以乐观，选不确定度大的项
4. 认为新 = 好，关键是定义“新”
5. 推断概率分布，依据分布“最佳”决策

探索方法的理论分析

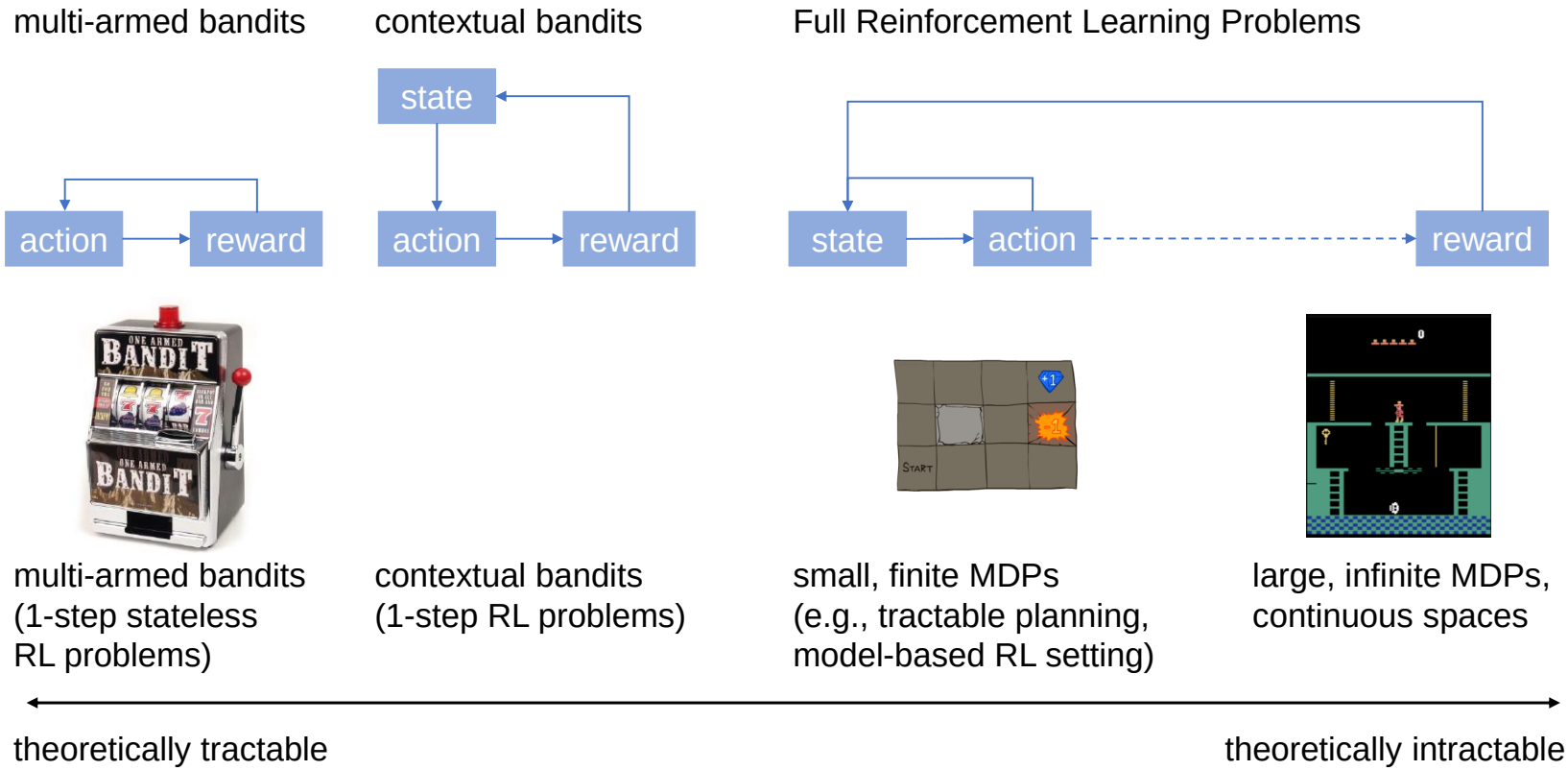


图 16: 探索问题的理论分析难度随状态空间规模而变化(Levine, 2018)

强化学习中的“探索与利用”方法

强化学习中的“探索与利用”方法

- 基于好奇心驱动的探索
- 基于熵正则的探索
- 用带噪声神经网络实现探索
- 高效利用样本经验
- 其他方法及理论工作

Intrinsic Curiosity Module(ICM)

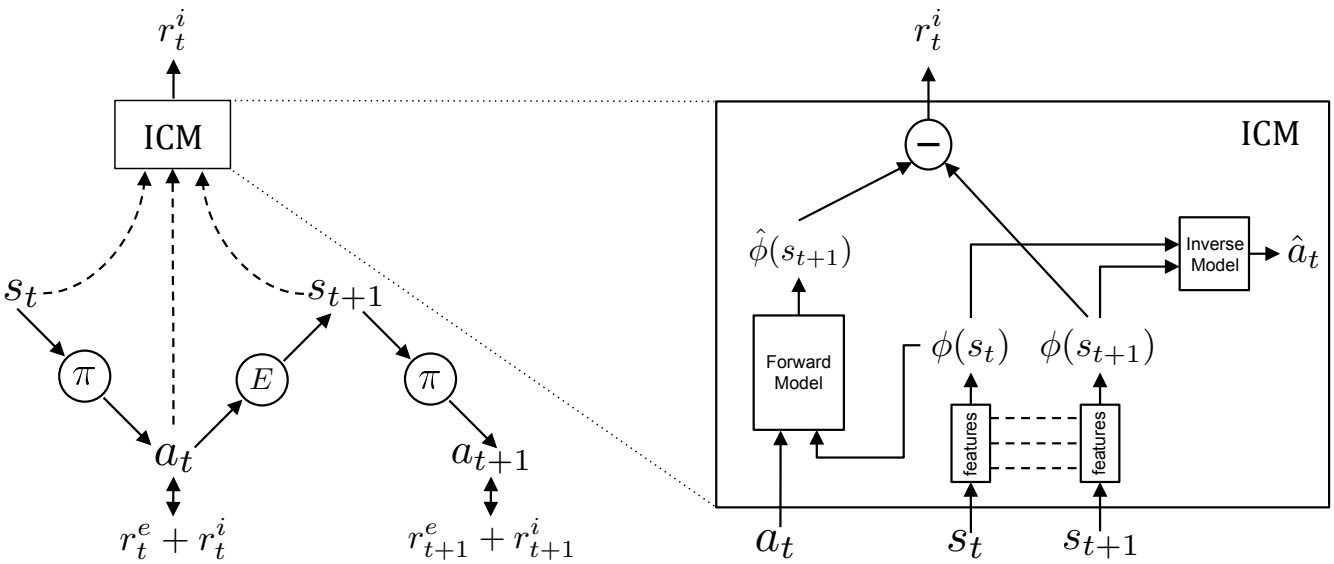


图 17: Intrinsic Curiosity Module(ICM)框架(Pathak et al., 2017)

$$L_F\left(\phi\left(s_t\right), \hat{\phi}\left(s_{t+1}\right)\right) = \frac{1}{2}\left\|f\left(\phi\left(s_t\right), a_t; \theta_F\right) - \phi\left(s_{t+1}\right)\right\|_2^2 \quad L_I\left(\hat{a}_t, a_t; \theta_I\right)$$

$$\min_{\theta_P, \theta_I, \theta_F}\left[-\lambda \mathbb{E}_{\pi\left(s_t; \theta_P\right)}\left[\Sigma_t r_t\right] + (1-\beta) L_I + \beta L_F\right]$$

Disagreement

- 多ICM集成，用模型间的不一致表达探索的额外收益

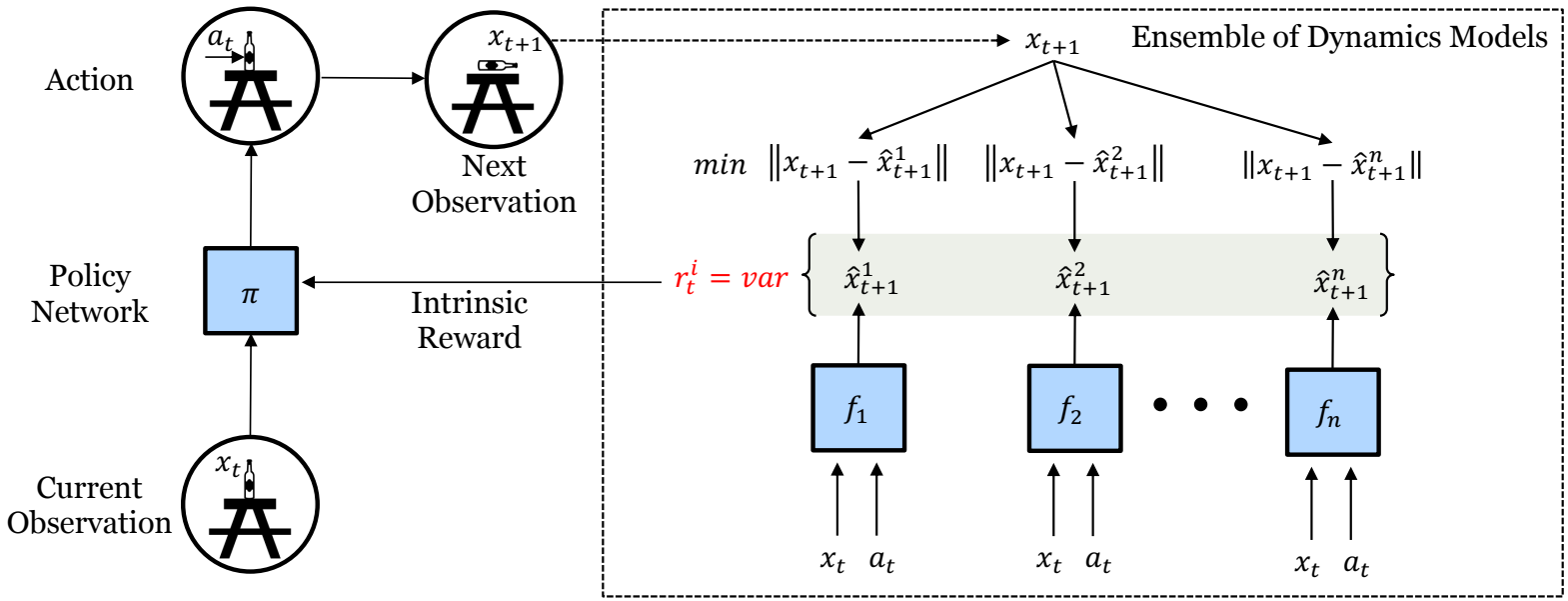


图 18: 基于不一致意见的探索模型框架(Pathak et al., 2019)

DORA

- 构造新MDP M'指引探索方向(Choshen et al., 2018)
  - 基于原MDP $M = (S, A, R, P, \gamma)$构造$M' = (S, A, 0, P, \gamma_E)$
  - 在M'上学习得到的Q值函数记做E值
  - 将M'中所有状态动作对的E值初始化为1，即$E(s, a) = 1$
  - 由于$E^*(s, a) = 0$，在训练过程中，状态动作对的$E(s, a)$将从1逐步变为0，接近真实的值函数

$$E(s, a) \leftarrow (1 - \alpha)E(s, a) + \alpha\left(0 + \gamma_E E\left(s', a'\right)\right)$$

  - 借助$\log_{1-\alpha} E$构造额外奖赏，指导动作选择

Episodic Curiosity

- 用可达性描述新颖度
- 用神经网络度量可达
- 基于可达定义增强奖赏

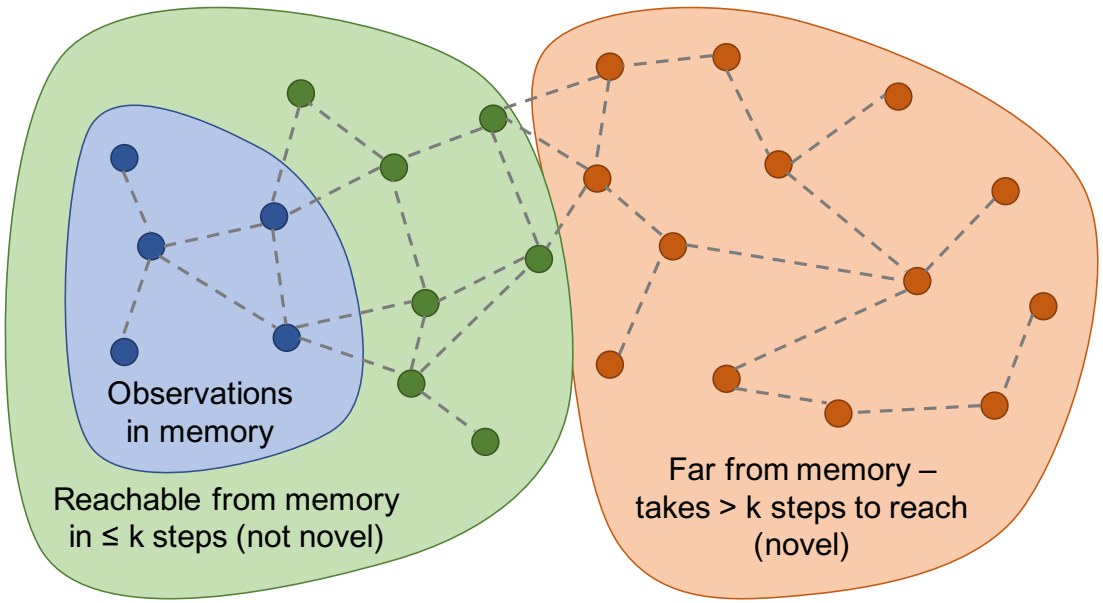


图 19: Episodic Curiosity中的新颖度(novelty)(Savinov et al., 2019)

Episodic Curiosity

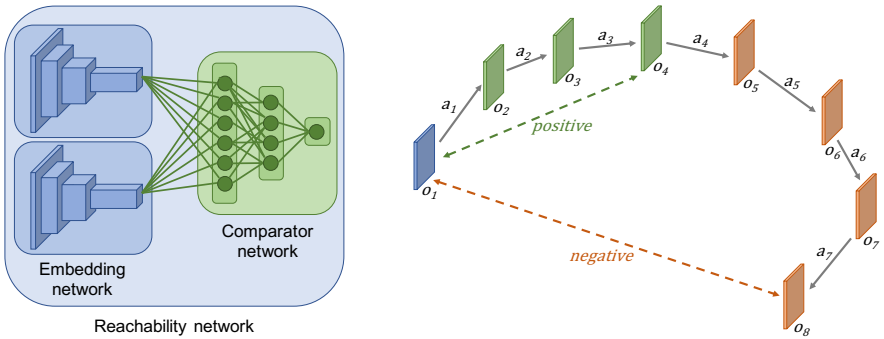


图 20: 可达性度量R-网络

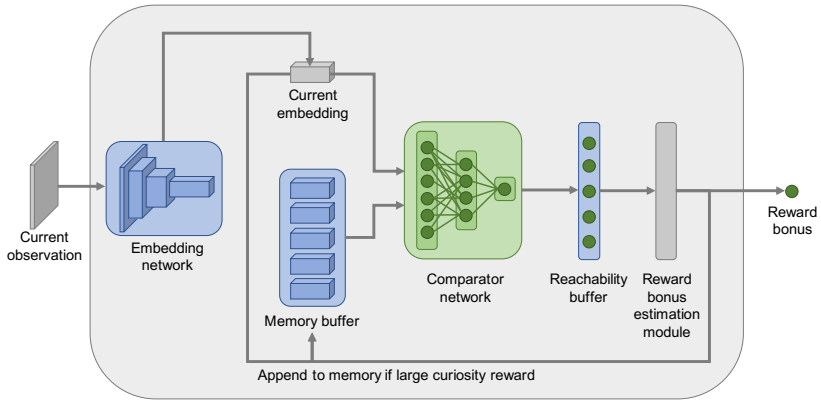


图 21: EC模型整体架构

$$\mathbf{e} = E(\mathbf{o}) \quad \text{embedding vector of } \mathbf{o}$$

$$\mathbf{M} = \langle \mathbf{e}_1, \ldots, \mathbf{e}_{|\mathbf{M}|} \rangle \quad \text{memory buffer stores } |\mathbf{M}| \text{ elements}$$

$$c_i = C(\mathbf{e}_i, \mathbf{e}), \quad i = 1, |\mathbf{M}| \quad \text{outputs of comparator network}$$

$$C(\mathbf{M}, \mathbf{e}) = F(c_1, \ldots, c_{|\mathbf{M}|}) \in [0, 1] \quad \text{aggregation function F, could be max}$$

$$b = B(\mathbf{M}, \mathbf{e}) = \alpha(\beta - C(\mathbf{M}, \mathbf{e})) \quad \text{bonus calculator}$$

Hash Based Pseudo-Counts

- 探索增强奖赏为$r^i : \mathcal{S} \mapsto \mathbb{R}$。

$$r^+(s) = \frac{\beta}{\sqrt{n(\phi(s))}}$$

- 通过$\phi(\mathbf{s})$将s压缩为k位编码，再计数$n(\phi(\mathbf{s}))$

$$\phi(s) = \text{sgn}(Ag(s)) \in \{-1, 1\}^k$$

- AutorEncoder编码更复杂或连续空间的观测

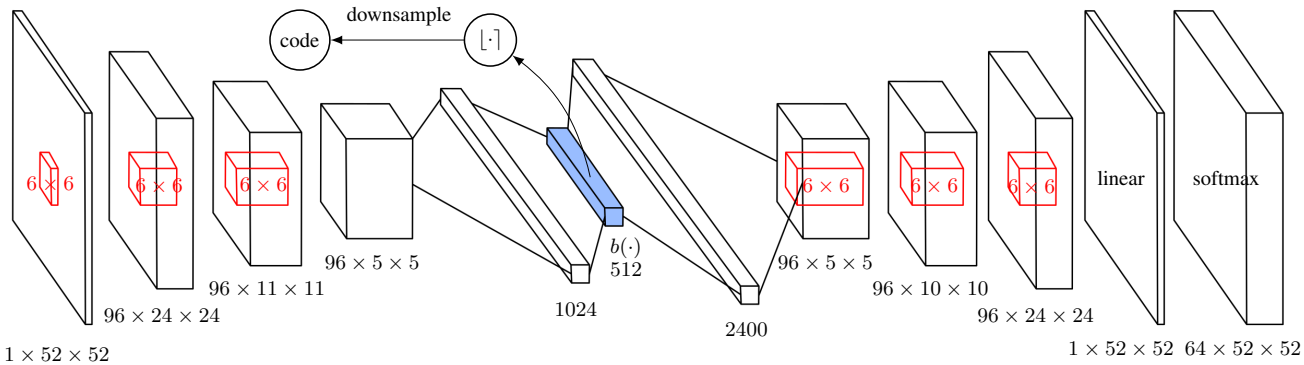


图 22: 基于自编码器的哈希编码计数方法架构(Tang et al., 2017)

Soft Q-Learning

- Q学习策略弊端
  - 连续状态、动作空间
  - 任务自适应性差

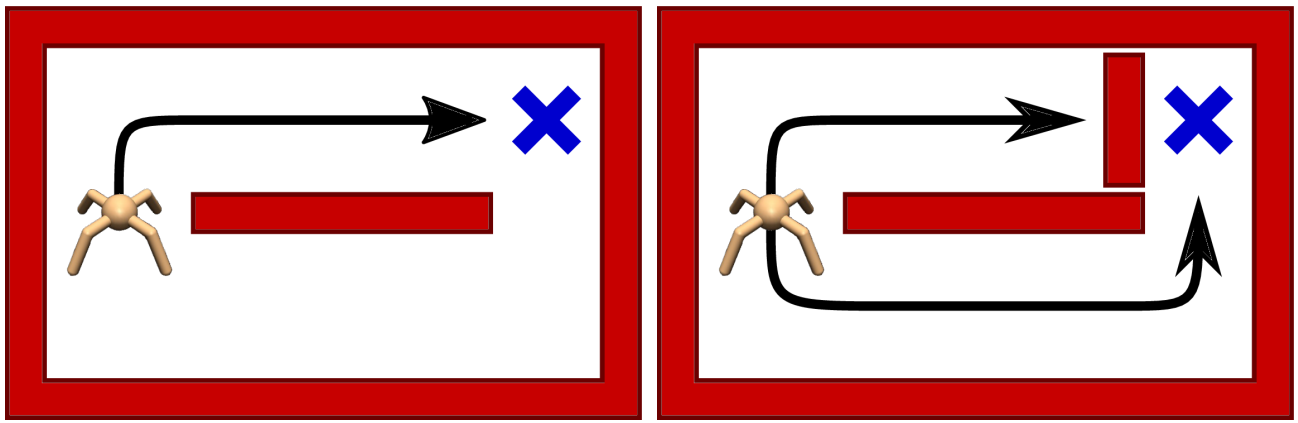


图 23: 巡航任务微调环境致最优策略彻底失效(Haarnoja et al., 2017)

Soft Q-Learning

- 引入熵正则化项并重构贝尔曼方程

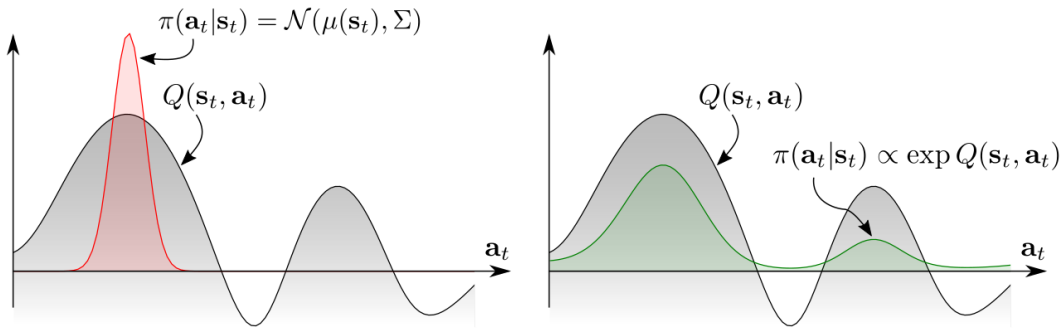


图 24: 将最优策略表达为玻尔兹曼分布

$$\pi^*_{\text{MaxEnt}} = \arg\max_{\pi} \sum_t \mathbb{E}_{(\mathbf{s}_t,\mathbf{a}_t)\sim\rho_\pi} \left[r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \alpha \mathcal{H}\left(\pi\left(\cdot|\mathbf{s}_t\right)\right)\right]$$

$$Q^*_{\text{soft}}\left(\mathbf{s}_t, \mathbf{a}_t\right) = r_t + \mathbb{E}_{(\mathbf{s}_{t+1},\ldots)\sim\rho_\pi} \left[\sum_{l=1}^{\infty} \gamma^l \left(r_{t+l} + \alpha \mathcal{H}\left(\pi^*_{\text{MaxEnt}}\left(\cdot|\mathbf{s}_{t+l}\right)\right)\right)\right]$$

$$V^*_{\text{soft}}\left(\mathbf{s}_t\right) = \alpha \log \int_{\mathcal{A}} \exp\left(\frac{1}{\alpha} Q^*_{\text{soft}}\left(\mathbf{s}_t, \mathbf{a}'\right)\right) d\mathbf{a}'$$

Soft Actor-Critic

- (Haarnoja et al., 2018)引入熵正则项，重构造贝尔曼方程
  - **策略评估**

$$\mathcal{T}^{\pi} Q\left(\mathbf{s}_t, \mathbf{a}_t\right) = r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \gamma \mathbb{E}_{\mathbf{s}_{t+1} \sim p}\left[V\left(\mathbf{s}_{t+1}\right)\right]$$

$$V\left(\mathbf{s}_t\right) = \mathbb{E}_{\mathbf{a}_t \sim \pi}\left[Q\left(\mathbf{s}_t, \mathbf{a}_t\right) - \log \pi\left(\mathbf{a}_t | \mathbf{s}_t\right)\right]$$

  - $Q^{k+1} = \mathcal{T}^{\pi} Q^{k}$反复迭代求得$Q$与$V$值收敛点
  - **策略提升**

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}}\left(\pi'\left(\cdot | \mathbf{s}_t\right) \| \frac{\exp\left(Q^{\pi_{\text{old}}}\left(\mathbf{s}_t, \cdot\right)\right)}{Z^{\pi_{\text{old}}}\left(\mathbf{s}_t\right)}\right)$$

  - 引入V值目标网络、采用重参数化技巧处理梯度更新

NoisyNet

- NoisyNet(Fortunato et al., 2018)在神经网络参数中掺杂噪声以增强策略探索性

$$y = wx + b$$

$$y \overset{\text{def}}{=} \left(\mu^w + \sigma^w \odot \varepsilon^w\right) x + \mu^b + \sigma^b \odot \varepsilon^b$$

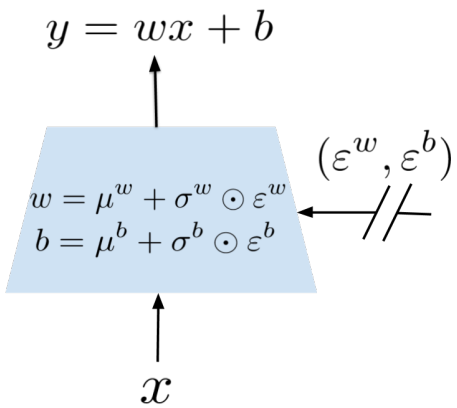


图 25: 引入随机噪声参数的线性层模型

Parameter Space Noise

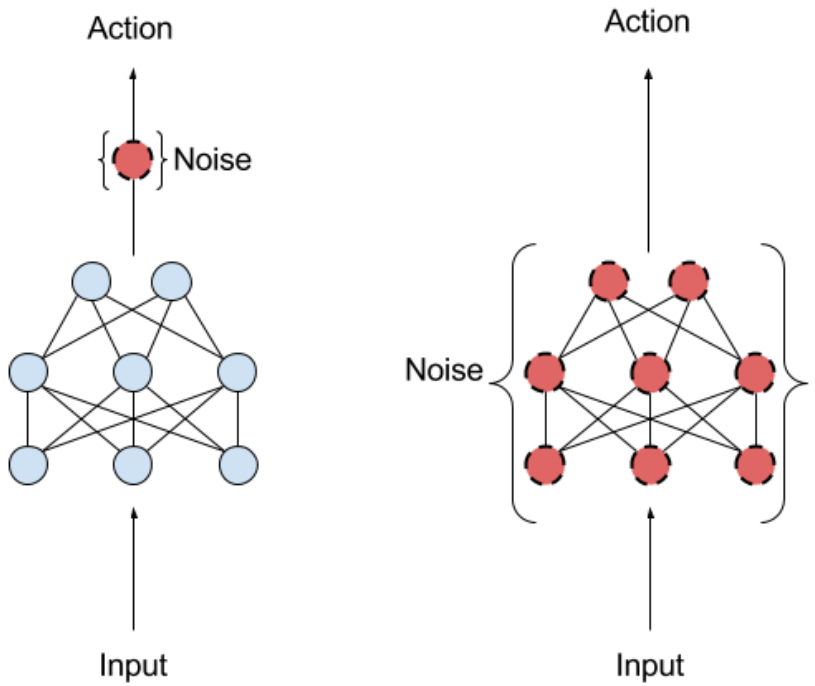


图 26: 动作空间和策略参数空间引入噪声对比(Plappert et al., 2017)

$$a_t = \pi(s_t) + \mathcal{N}\left(0, \sigma^2 I\right)$$

$$\widetilde{\theta} = \theta + \mathcal{N}\left(0, \sigma^2 I\right)$$

RND

- RND(Burda et al., 2019)引入网络蒸馏技术，Target Network有固定的随机初始化的权重，在训练中不改变，它能够为每一个状态观测输出一个特征表示
- Predictor Network将Target Network的输出结果作为label 并根据MSE损失更新参数θ

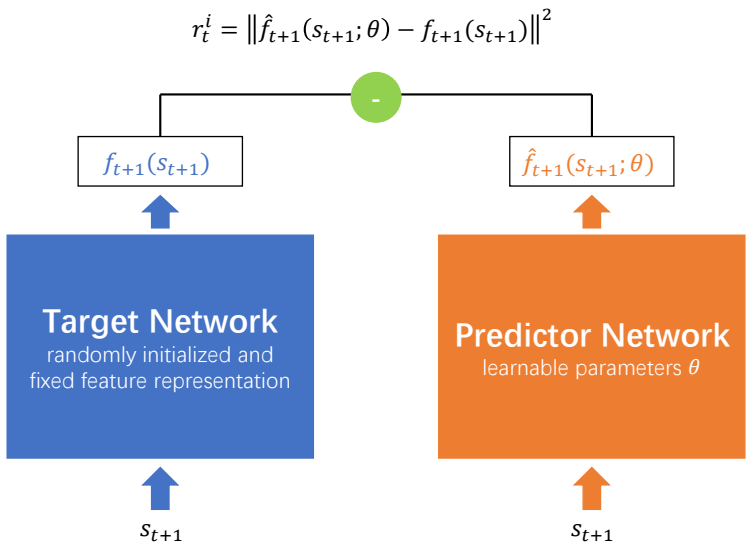


图 27: Random Network Distillation(RND)模型产生内在增强奖赏

RND

- 与ICM不同，RND框架中前向网络易于实现和计算，不以s_t,a_t作输入预测下一状态特征表示

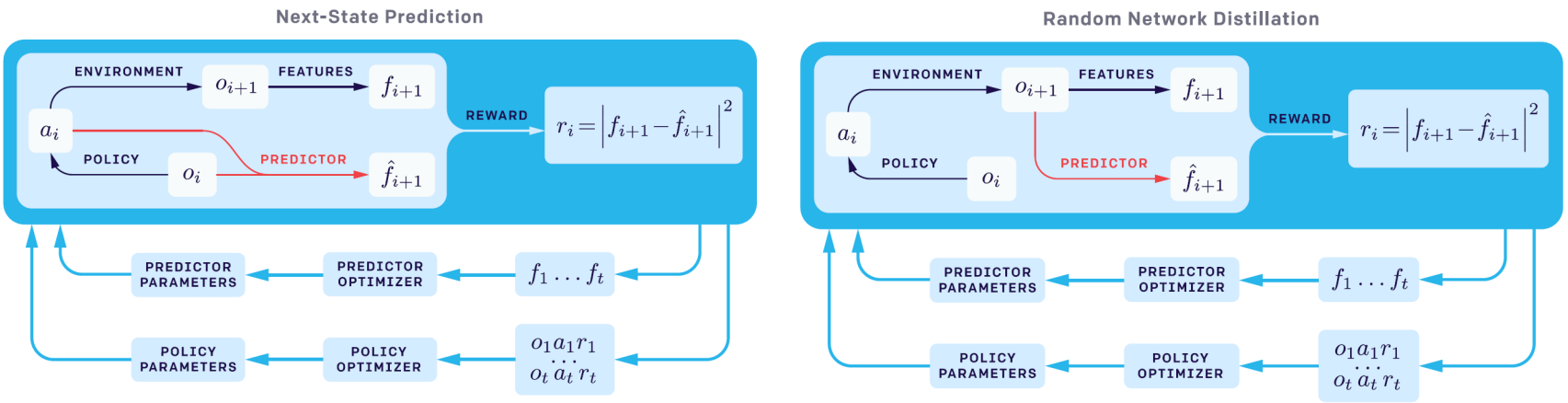


图 28: RND与ICM中前向预测模块对比

Hindsight Experience Replay

- HER(Andrychowicz et al., 2017)解决稀疏的二元奖赏任务
  - 预设目标g，轨迹序列为$s_0, s_1, s_2, \cdots, s_T \neq g$，将目标重设为$s_T$，未成功完成任务变为成功
  - 扩展目标集合$\mathcal{G}(g \in \mathcal{G})$，有利于扩充经验池中正例采样轨迹

$$a_t \leftarrow \pi_b\left(s_t \| g\right) \quad \|\text{意味着将}s_t\text{和}g\text{串接起来}$$

$$r_t := r\left(s_t, a_t, g\right) \quad g\text{随机取自}\mathcal{G}$$

$$\text{Replay Buffer} \leftarrow \left(s_t \| g, a_t, r_t, s_{t+1} \| g\right) \quad \text{串接}s_t\text{和}g\text{存入经验回放池}$$

- 课程学习(Curriculum Learning)(Bengio et al., 2009)

Go-Explore

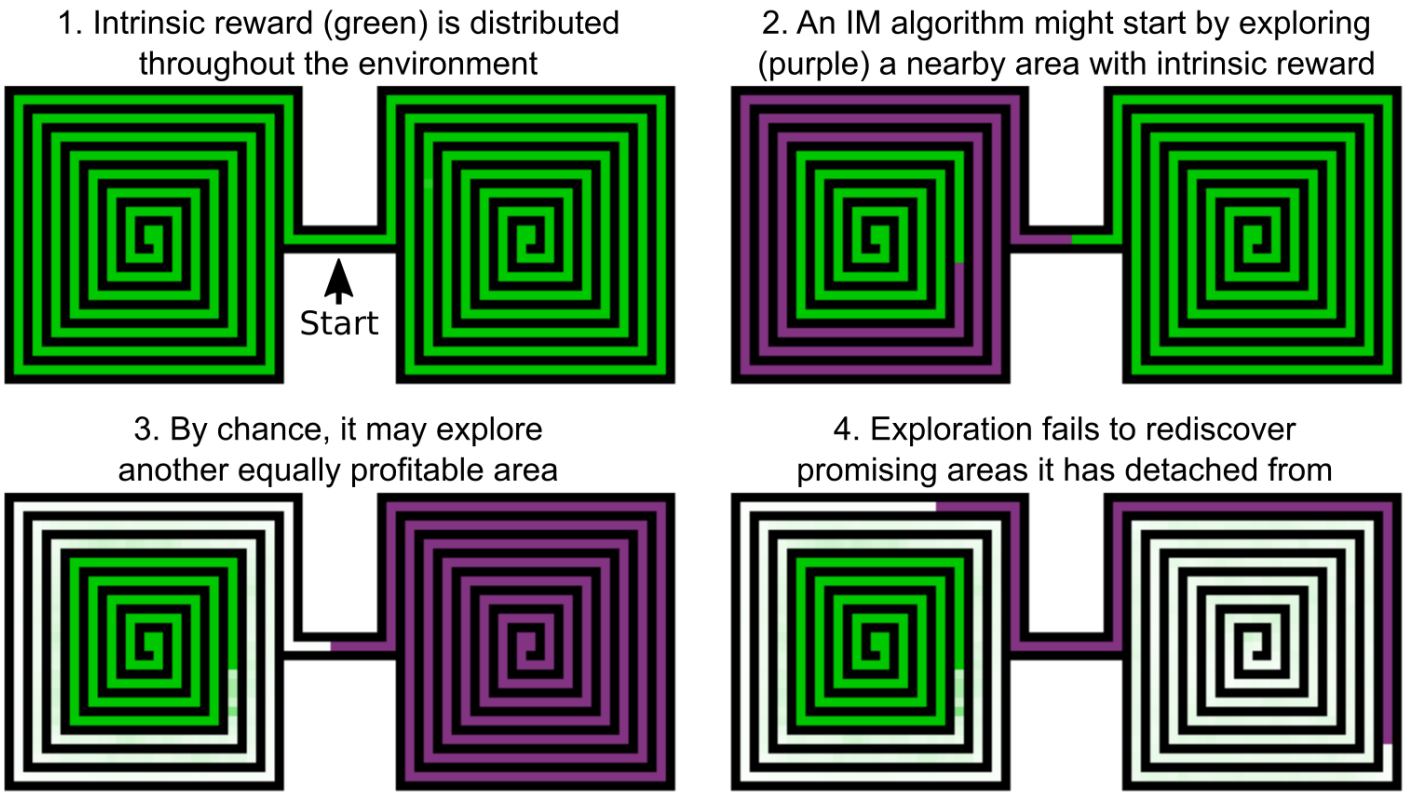


图 29: Intrinsic Motivation(IM)类方法存在的问题

Go-Explore

- Uber提出Go-Explore解决门特祖玛复仇问题
  - 存档探索过的路径
  - 如有必要对存档路径模仿学习
  - 图像粗粒化

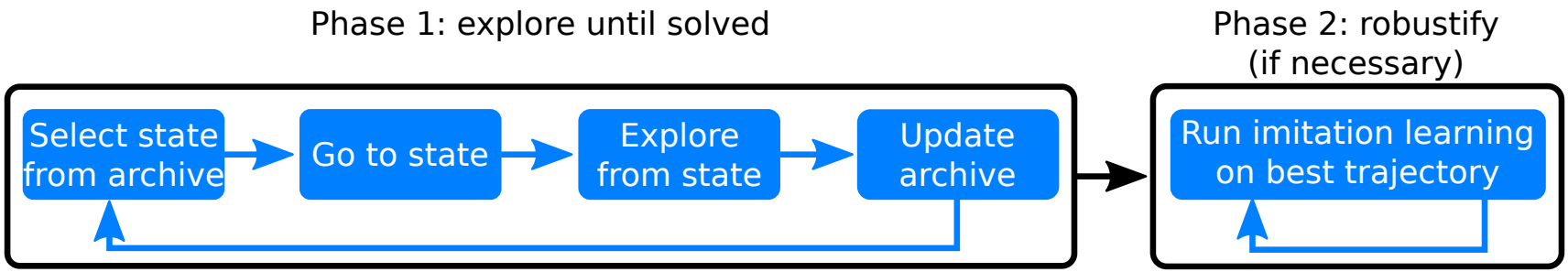
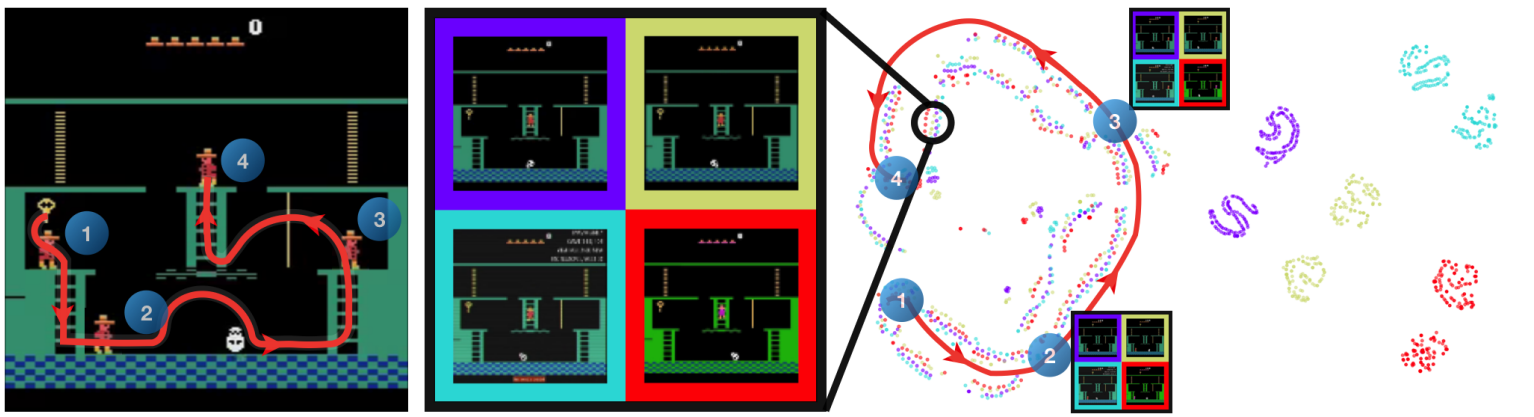


图 30: Go-Explore方法(Ecoffet et al., 2019)

TDC+CMC

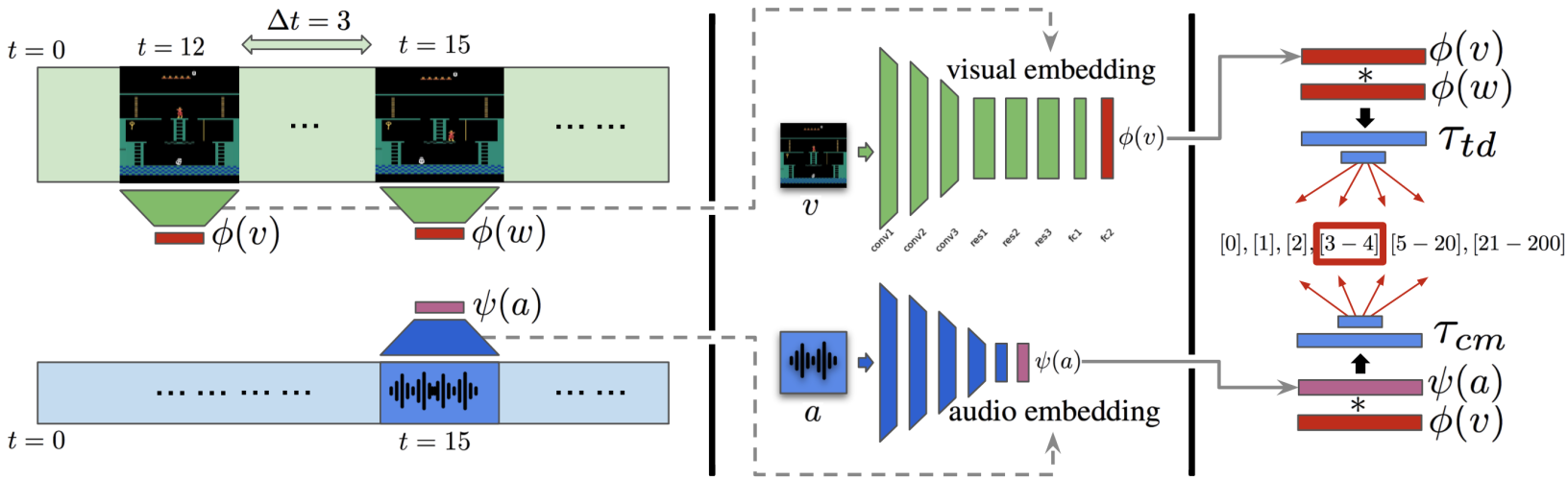
- Google团队利用人类玩家玩蒙特祖玛复仇的YouTube多版本在线录像来训练AI(Aytar et al., 2018)
  - 自监督学习方式对齐不同录像
  - TDC 画面时间差预测(**T**emporal **d**istance **c**lassification)
  - CMC 画面与声音对齐的跨模态时间差预测(**C**ross-**m**odal temporal distance **c**lassification)



(a) An example path (b) Aligned frames (c) Our embedding (d) Pixel embedding

图 31: 学习表征将不同画面在表征空间中对齐

TDC+CMC



(a) Temporal and cross-modal pair selection (b) Embedding networks (c) Classification networks

图 32: 基于YouTube示例录像的模仿学习神经网络模型

LfSD

- LfSD(Salimans and Chen, 2018)不断重新设置出发点，将长期规划任务拆分成多个子任务

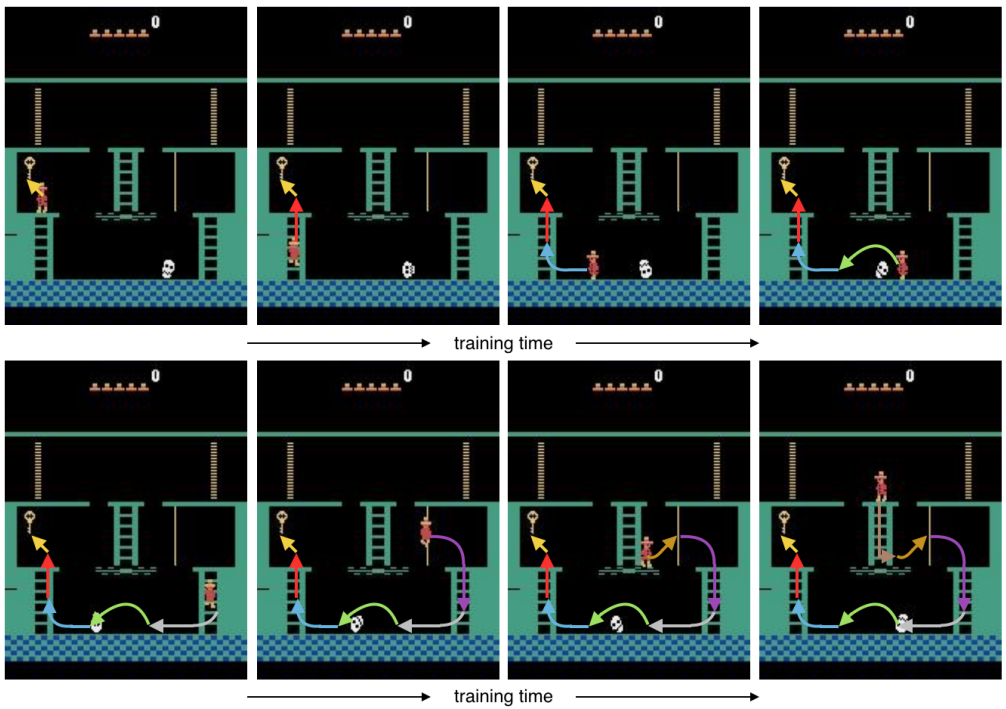


图 33: 逐步后置起始点从而引导智能体学习到更长的行动路径

R2D3

- R2D3(Paine et al., 2020)
  - 多个actor进程并行采样
  - 所有actor进程共享经验缓冲池
  - 全局learner运行Double DQN网络
  - 示例经验和actor采样经验按比例(ρ)混合

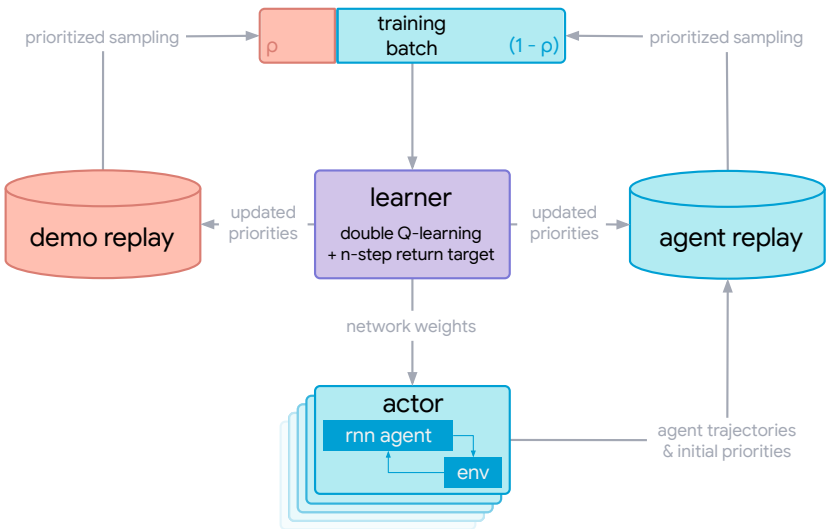


图 34: Recurrent Replay Distributed DQN from Demonstrations(R2D3)框架

Bootstrapped DQN

- Bootstrapped DQN(Osband et al., 2016)
  - 同时学习关于(s, a)的$K \in \mathbb{N}$个Q值来近似建模出Q值分布
  - 多个Q值网络共享一部分参数
  - 动作选取呈现出受Q值分布影响的探索效果

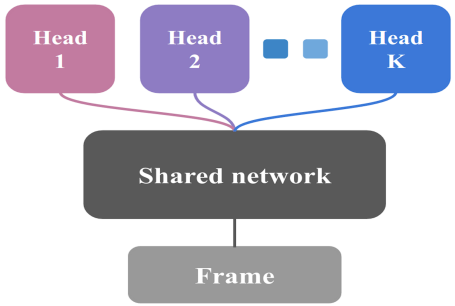


图 35: Bootstrapped DQN架构

VIME

- VIME(Houthooft et al., 2016)是基于变分信息最大化的探索方法
- 智能体连续决策并降低环境的不确定性，形式化为最大化在$\{a_t\}$上的累计熵减

$$\sum_t \left(H\left(\theta|\xi_t, a_t\right) - H\left(\theta|s_{t+1}, \xi_t, a_t\right)\right)$$

- 定义增强奖赏

$$\eta D_{\mathrm{KL}}\left[p\left(\theta|\xi_t, a_t, s_{t+1}\right) \| p\left(\theta|\xi_t\right)\right]$$

DQN-CTS

- UCB类方法中引入状态计数或状态-动作对的计数$\hat{N}(s)$度量不确定性，并依据计数构造增强奖赏$\mathcal{B}(\hat{N}(s))$

$$r^{+}(s) = r_i(s) + \mathcal{B}(\hat{N}(s))$$

$$\sqrt{\frac{2\ln T}{\hat{N}(s)}}$$

UCB1

(Auer et al., 2002)

$$\sqrt{\frac{1}{\hat{N}(s)}}$$

MBIE-EB

(Strehl and Littman, 2008)

$$\frac{1}{\hat{N}(s)}$$

BEB

(Kolter and Ng, 2009)

DQN-CTS

- (Bellemare et al., 2016a)证明虚拟计数$\hat{N}_n(x)$和IG与PG之间的关系

$$\hat{N}_n(x) \approx \left(e^{\mathrm{PG}_n(x)} - 1\right)^{-1}$$

$$IG_n(x) \leq PG_n(x) \leq \hat{N}_n(x)^{-1}$$

$$PG_n(x) \leq \hat{N}_n(x)^{-1/2}$$

- Information Gain(IG)

$$\mathrm{IG}_n(x) := \mathrm{IG}\left(x; x_{1:n}\right) := \mathrm{KL}\left(w_n(\cdot, x) \| w_n\right)$$

- Prediction Gain(PG)

$$\mathrm{PG}_n(x) := \log \rho'_n(x) - \log \rho_n(x)$$

DQN-PixelCNN

- (Ostrovski et al., 2017)以Pixel-CNN作神经网络密度模型
- 定义增强奖赏

$$r^+(x) := \left(\hat{\mathrm{N}}_n(x)\right)^{-1/2}$$

$$\hat{\mathrm{N}}_n(x) \approx \left(e^{\mathrm{PG}_n(x)} - 1\right)^{-1}$$

总结

Montezuma's Revenge

算法	文献	Montezuma's Revenge
SARSA	(Mnih et al., 2015)(Bellemare et al., 2012)	259
DQN	(Mnih et al., 2015)(Wang et al., 2016)	0
DDQN	(van Hasselt et al., 2016)(Wang et al., 2016)	42
Duel. DQN	(Wang et al., 2016)	22
Prior. DQN	(Schaul et al., 2016)	13
A3C	(Mnih et al., 2016)	67
DQN-CTS	(Bellemare et al., 2016b)	3705
DQN-PixelCNN	(Ostrovski et al., 2017)	2514
Rainbow	(Hessel et al., 2018)	154
RND	(Burda et al., 2019)	11347
Go-Explore	(Ecoffet et al., 2019)	43763
Go-Explore (domain knowledge)	(Ecoffet et al., 2019)	666474
Go-Explore (best)	(Ecoffet et al., 2019)	18003200
LfSD (best)	(Salimans and Chen, 2018)	74500
TDC+CMC	(Aytar et al., 2018)	41098
Average Human	(Pohlen et al., 2018)	4753
Human Expert	(Pohlen et al., 2018)	34900
Human World Record	(ataricompendium)	1219200

总结

- 深度强化学习“探索与利用”方法应用
- 从优化角度深入理解策略梯度类算法的探索技巧

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen. King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *CoRR*, abs/1511.05952, 2016.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. *ArXiv*, abs/1511.06581, 2016.

Matteo Hessel, Joseph Modayil, H. V. Hasselt, T. Schaul, Georg Ostrovski, W. Dabney, Dan Horgan, B. Piot, Mohammad Gheshlaghi Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. *ArXiv*, abs/1710.02298, 2018.

Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 1057–1063. MIT Press, 2000. URL http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf.

Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

David Silver, Guy Lever, Nicolas Manfred Otto Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

Ron Amit, R. Meir, and K. Ciosek. Discount factor as a regularizer in reinforcement learning. *ArXiv*, abs/2007.02040, 2020.

Silviu Pitis. Rethinking the discount factor in reinforcement learning: A decision theoretic approach. *ArXiv*, abs/1902.02893, 2019.

M. Littman and Csaba Szepesvari. A generalized reinforcement-learning model: Convergence and applications. In *ICML*, 1996.

Kavosh Asadi and M. Littman. An alternative softmax operator for reinforcement learning. In *ICML*, 2017.

L. Pan, Qingpeng Cai, Q. Meng, Wei Chen, Longbo Huang, and T. Liu. Reinforcement learning with dynamic boltzmann softmax updates. *ArXiv*, abs/1903.05926, 2019.

Yuhuai Wu, Elman Mansimov, Roger B. Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. *ArXiv*, abs/1708.05144, 2017.

D. Silver, Julian Schrittwieser, K. Simonyan, Ioannis Antonoglou, Aja Huang, A. Guez, T. Hubert, L. Baker, Matthew Lai, A. Bolton, Yutian Chen, T. Lillicrap, F. Hui, L. Sifre, George van den Driessche, T. Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017.

Yuri Burda, Harrison A Edwards, Amos J. Storkey, and Oleg Klimov. Exploration by random network distillation. *ArXiv*, abs/1810.12894, 2019.

Shuyuan He. *Stochastic Process(in Chinese)*. Peking University Press, 2008.

Conglu Kang. *Theory and Applications of Monte Carlo Methods(in Chinese)*. Science Press, 2015.

Baorong Fang, Jidong Zhou, and Yimin Li. *Matrix(in Chinese)*. Tsinghua University Press, 2013.

George Casella and Roger L. Berger. Statistical inference second edition. 2007.

Allen B. Downey. *Think Bayes: Bayesian Statistics in Python*. O'Reilly Media, 2013.

Jeremy Orloff and Jonathan Bloom. Bayesian updating with continuous priors. https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading13a.pdf.

John Duchi. Supplemental lecture notes hoeffding's inequality. http://cs229.stanford.edu/extra-notes/hoeffding.pdf.

C Q., Thomas M. Cover, and Joy A. Thomas. Elements of information theory. *Publications of the American Statal Association*, 103(481):429–429, 2006.

Li Gao. *Numerical Optimization Method(in Chinese)*. Peking University Press, 2014.

Sergey Levine. Deep reinforcement learning. http://rail.eecs.berkeley.edu/deeprlcourse-fa18/, 2018.

Liangpeng Zhang. *Sample Efficiency in Reinforcement Learning*. PhD thesis, 2019.

Shivani Agarwal. Stochastic multi armed bandits. https://www.shivani-agarwal.net/Teaching/E0370/Aug-2013/Lectures/22.pdf, 2013.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47: 235–256, 2002.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *NIPS*, 2011.

J.T. Vanderplas, A.J. Connolly, Ž. Ivezić, and A. Gray. Introduction to astroml: Machine learning for astrophysics. In *Conference on Intelligent Data Understanding (CIDU)*, pages 47 –54, oct. 2012. doi: 10.1109/CIDU.2012.6382200.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *ArXiv*, abs/1403.5556, 2014.

Lilian Weng. Exploration strategies in deep reinforcement learning. *lilianweng.github.io/lil-log*, 2020. URL https://lilianweng.github.io/lil-log/2020/06/07/exploration-strategies-in-deep-reinforcement-learning.html.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 488–489, 2017.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement, 2019.

Leshem Choshen, Lior Fox, and Yonatan Loewenstein. Dora the explorer: Directed outreaching reinforcement action-selection. *ArXiv*, abs/1804.04012, 2018.

Nikolay Savinov, Anton Raichuk, Raphael Marinier, Damien Vincent, Marc Pollefeys, Timothy P. Lillicrap, and Sylvain Gelly. Episodic curiosity through reachability. *ArXiv*, abs/1810.02274, 2019.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Exploration: A study of count-based exploration for deep reinforcement learning. *ArXiv*, abs/1611.04717, 2017.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *ArXiv*, abs/1702.08165, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Rémi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. *ArXiv*, abs/1706.10295, 2018.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. *ArXiv*, abs/1706.01905, 2017.

Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel H Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *ArXiv*, abs/1707.01495, 2017.

Yoshua Bengio, Jér?me Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, 2009.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Go-explore: a new approach for hard-exploration problems. *ArXiv*, abs/1901.10995, 2019.

Yusuf Aytar, Tobias Pfaff, David Budden, Thomas Paine, Ziyu Wang, and Nando de Freitas. Playing hard exploration games by watching youtube. In *NeurIPS*, 2018.

Tim Salimans and Richard Chen. Learning montezuma's revenge from a single demonstration. *ArXiv*, abs/1812.03381, 2018.

Tom Le Paine, Caglar Gulcehre, Bobak Shahriari, Misha Denil, Matthew D. Hoffman, Hubert Soyer, Richard Tanburn, Steven Kapturowski, Neil C. Rabinowitz, D.V.J. Williams, Gabriel Barth-Maron, Ziyu Wang, Nando de Freitas, and Worlds Team. Making efficient use of demonstrations to solve hard exploration problems. *ArXiv*, abs/1909.01387, 2020.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in neural information processing systems*, pages 4026–4034, 2016.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *NIPS*, 2016.

Alexander L. Strehl and M. Littman. An analysis of model-based interval estimation for markov decision processes. *J. Comput. Syst. Sci.*, 74:1309–1331, 2008.

J. Z. Kolter and A. Ng. Regularization and feature selection in least-squares temporal difference learning. In *ICML '09*, 2009.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016a.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pages 2721–2730, 2017.

Marc G. Bellemare, J. Veness, and Michael Bowling. Investigating contingency awareness using atari 2600 games. In *AAAI*, 2012.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *NIPS*, 2016b.

Tobias Pohlen, B. Piot, T. Hester, Mohammad Gheshlaghi Azar, Dan Horgan, D. Budden, Gabriel Barth-Maron, H. V. Hasselt, John Quan, Mel Vecerík, Matteo Hessel, R. Munos, and Olivier Pietquin. Observe and look further: Achieving consistent performance on atari. *ArXiv*, abs/1805.11593, 2018.

ataricompendium. Atari vcs/2600 scoreboard. http://www.ataricompendium.com/game_library/high_scores/high_scores.html.