Efficient Reinforcement Learning with a Mind-Game for Full-Length StarCraft II

Ruo-Ze Liu, Haifeng Guo, Xiaozhong Ji, Yang Yu†, Zitai Xiao,
Yuzhou Wu, Zhen-Jia Pang, Tong Lu
National Key Laboratory for Novel Software Technology, Nanjing University, China
†Correspondence: yuy@nju.edu.cn

【强化学习 72】MindGame

张楚珩
清华大学 交叉信息院博士在读

原文传送门

Liu, Ruo-Ze, et al. "Efficient Reinforcement Learning with a Mind-Game for Full-Length StarCraft II." arXiv preprint arXiv:1903.00715 (2019).

特色

为了求解复杂MDP，有两个最为重要的方式：state abstraction，把庞大的状态空间通过某种映射抽象为较小的状态空间，这样要学习到该MDP上的一个较好的策略所需要的样本数就会更少；temporal abstraction，把时序上若干个有意义的行动抽象为一个高级的行动，这样状态空间能够更加有效地被探索，探索到目标状态的平均所需样本数就会更少。总体说来，当一个MDP很困难的时候，我们希望找到一个abstraction能够把原本困难的MDP压缩为一个简单的MDP。

这篇文章某种程度上可以说是反过来做这件事情，就是人工设计一个简单的MDP，反映原MDP的实质但是容易被解决，然后把简单MDP上学习到的策略迁移到去解决原MDP。人工智能的终极目标肯定不需要人工设计这样的简单的MDP，但是这确是一个工业上解决复杂强化学习问题一种有效的解决方案，也为之后的工作提供了很多启发。

过程

1. 整体过程

先放一张图，具体的后面来解释。

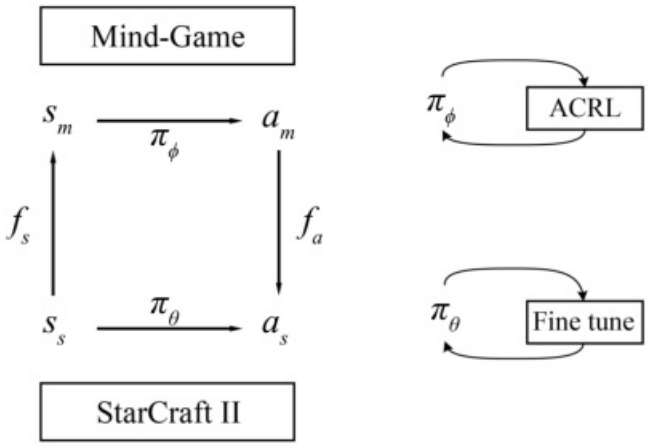

Figure 1: Mind-Game Architecture.

2. 抽象MDP

假设原MDP为 $M_s = < S_s, A_s, P_s(\cdot), R_s(\cdot), \gamma, T_s >$ ，其中 T_s 表示一个回合的最大步数。考虑一个抽象的MDP为

$$M_m = \langle S_m, A_m, P_m(\cdot), R_m(\cdot), \gamma, T_m \rangle \tag{3}$$

如果学习到了一个抽象MDP上的策略 π_ϕ ，并且假设抽象的MDP能够反映原MDP的实质，给定两者之间的一些映射关系之后，可以写出原MDP上的策略表示

$$\pi_\theta(s_s) = f_a(\pi_\phi(f_s(s_s))) \tag{5}$$

其中 $f_s(\cdot)$ 表示原状态往抽象状态上的映射， $f_a(\cdot)$ 表示抽象的动作往原动作上的映射。它们之间的关系可以见前面的那张图。

显而易见，这样的映射表示也是很难表示出来或者学习到的，因此文章对于抽象的MDP和原MDP之间的关系做了一定的限制。

$$\begin{aligned} M_m = \langle & S_m \subset S_s, A_m = A_s, \\ & P_m(\cdot) \approx P_s(\cdot), \\ & R_m(\cdot) = R_s(\cdot), \gamma_m = \gamma_s, T_m \rangle \end{aligned} \tag{6}$$

由于抽象MDP中状态空间和原MDP中相同，同时行动空间也一样，这样学习到了一个抽象MDP上的策略 π_ϕ 之后就能够之间用到原MDP上。

3. 学习过程

学习过程分为两个阶段。

在第一个阶段中，智能体在抽象的MDP中进行学习。

抽象的MDP是人工写的程序。其状态空间和原MDP维度相同（我猜想，应该也是渲染出来的游戏界面截图吧，文章好像没说）。其行动空间和原MDP相同。其状态转移是人为写的，个人感觉就是写了一个和原来StarCraftII原则上一样，但是基于回合制的（即时间尺度上可能不太一样）游戏以及基于脚本的不同难度的对战Bots。做这篇工作的同学真的是非常热爱StarCraftII这款游戏了，他写的这个mindgame考虑了建筑的建造费用、血量、防御、建造时间这一类的有准确数据的参数（游戏wiki上面应该都有），同时也近似估计了资源收集、建造单位等相关的参数等。抽象MDP中的奖励函数还是使用最终的获胜条件 $\{-1,0,+1\}$ ，如果使用更为密集的奖励函数，学习会变得不太稳定。

由于写了参数化难度可调的对战bots，因此可以拿来做自适应课程学习（adaptive curriculum reinforcement learning，ACRL），即先在比较简单的环境中训练，如果获胜率超过一定的阈值那么就进入下一个难度等级进行训练。

在第二个阶段中，智能体在原MDP中进行学习。

由于特殊的设定，抽象空间中学习到的策略能够直接用在原MDP上，但是效果不会太好，因此再在原MDP上进一步地进行学习，学习方法和他们之前工作（【强化学习算法 16】NJUStarCraft）的方法类似。

实验

实验效果如下图所示，左图红线为ACRL在mindgame中学习得到的学习曲线，右图为在原任务上学习的学习曲线，纵坐标为胜率。可以看到mindgame中学习得到的曲线最后基本上达到了90%以上的胜率，直接迁移到原MDP上之后一开始的胜率只有50%左右，不过很快就能够学习到90%左右的胜率水平。值得一提的是，他们之前的工作需要在单机上训练一天，现在只需要一小时。

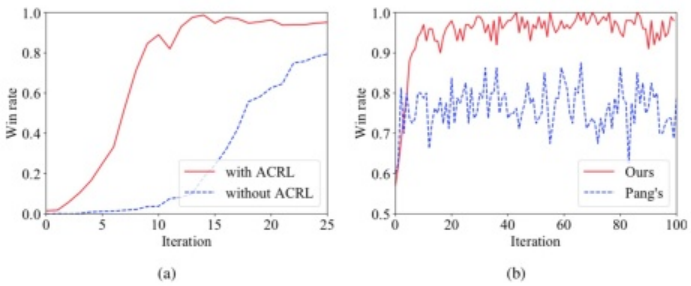

Figure 2: (a) The process of training the agent in mind-game with ACRL. (b) Transfer learning on *Simple64*.

发布于 2019-06-17

强化学习 (Reinforcement Learning)

文章被以下专栏收录

强化学习前沿
读呀读paper

进入专栏