

# Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak <sup>1</sup> Pulkit Agrawal <sup>1</sup> Alexei A. Efros <sup>1</sup> Trevor Darrell <sup>1</sup>

# 【强化学习算法 23】ICM



张楚珩 💙

清华大学 交叉信息院博士在读

4 人赞同了该文章

这篇工作可以认为是本专栏讲到的另一篇工作<u>【强化学习算法 17】Curiosity</u> 的后续,名字是 Intrinsic Curiosity Module的缩写。

### 原文传送门:

Burda, Yuri, et al. "Large-scale study of curiosity-driven learning." arXiv preprint arXiv:1808.04355 (2018).(前面那篇工作)

Pathak, Deepak, et al. "Curiosity-driven exploration by self-supervised prediction." International Conference on Machine Learning (ICML). Vol. 2017. (本文讲的工作)

Zhelo, Oleksii, et al. "Curiosity-driven Exploration for Mapless Navigation with Deep Reinforcement Learning." arXiv preprint arXiv:1804.00456 (2018). (这篇工作的一个应用: 在寻路方面的一个应用)

### 特色:

参看本专栏前面讲的那篇文章,curiosity这种intrinsic reward是基于智能体对下一步预测与实际下一步状态差距来定义的。不过之前的文章提到这种定义会出现的一个问题,那就是当环境出现与智能体无关的随机性的时候,智能体会因为始终不能预测下一步的状态,而卡在相应的位置。这篇文章就解决了这个问题。

同时,这篇文章还通过测试算法transfer learning的性能来研究其泛化性能。

#### 过程:

### 1. 通常的curiosity定义

一般来说,会先维护一个前向模型(forward model)来预测下一步的状态

$$\hat{\phi}(s_{t+1}) = f(\phi(s_t), a_t; \theta_F)$$
(4)

其中,  $\phi(s_i)$  是对于当前状态的表示,这个表示在有的文章里面就直接用原来的状态空间,即  $\phi(s)=s$ ,有的会对于表示进行学习。这里定义它为一个神经网络  $\phi(s_i)=\phi(s_i;\theta_s)$ 。

curiosity就定义为实际状态表示和前向模型求得状态表示的差距

$$r_t^i = \frac{\eta}{2} \|\hat{\phi}(s_{t+1}) - \phi(s_{t+1})\|_2^2$$
 (6)

而前向模型和表示参数的学习优化如下回归误差得到

$$\min_{\theta_F,\theta_E} L_F\Big(\hat{\phi}(s_{t+1}), \phi(s_{t+1})\Big) \tag{5}$$

#### 2. 反向模型的学习

从以上的公式中我们可以看到,当环境中出现与智能体无关的随机因素的时候,无论前向模型学习地多好,都不可能准确地预测出来下一个状态。因此,这种情况下会始终有一个较大的奖励,从而使得智能体"卡在"这个位置。这就是【强化学习算法 17】Curiosity中提到的被随机电视画面卡住的问题。为了解决这个问题,我们来分析一下状态包含一些什么信息。

- 1. 可以被智能体控制的部分;
- 2. 不能被智能体控制但是可以影响智能体的部分;
- 3. 既不能被控制也不能影响智能体的部分;
- 一个好的表示应该包含前两项的信息,而不包含后两项的信息。文中用了一个巧妙的办法来做到这一点,就是另外学习一个**反向模型(inverse model)**。

$$\hat{a}_t = g\left(\phi(s_t), \phi(s_{t+1}); \theta_I\right) \tag{2}$$

反向模型和状态表示的参数都通过以下优化来更新

$$\min_{\theta_I,\theta_E} L_I(\hat{a}_t, a_t) \tag{3}$$

为了学习到一个好的反向模型,其表示自然会不去包含与其无关的内容,因此上面描述的第三点就不会被包含在内了。

#### 3. 总体流程

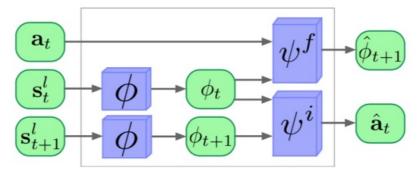


Fig. 2: ICM architecture.  $\mathbf{s}_t^l$  and  $\mathbf{s}_{t+1}^l$  are first passed through the feature extraction layers  $\phi$ , and encoded into  $\phi_t$  and  $\phi_{t+1}$ . Then  $\phi_t$  and  $\phi_{t+1}$  are input together into the *inverse model*  $\psi^i$ , to infer the action  $\hat{\mathbf{a}}_t$ . At the same time,  $\mathbf{a}_t$  and  $\phi_t$  are together used to predict  $\hat{\phi}_{t+1}$ , through the *forward model*  $\psi^f$ . The prediction error between  $\hat{\phi}_{t+1}$  and  $\phi_{t+1}$  is used as the intrinsic reward  $R^i$ .

ICM流程框图(为了展示清晰,使用了文献[3]中的图,而不是原文中的图)

# 4. 实验

- 在VizDoom环境中在原本稀疏的奖励上加上这里定义的curiosity奖励能够学习地更好;
- 把只使用curiosity训练的智能体扔到新环境中看看能不能有好的表现,实验显示,在简单环境中 学习好的智能体能够更快更好地再更复杂的环境中学习,这表面curiosity确实能够使得智能体拥 有泛化能力;

实验中的一个问题

Level Ids Accuracy Iterations	Level-1 Scratch 1.5M	Level-2				Level-3			
		Run as is	Fine-tuned 1.5M	Scratch 1.5M	Scratch 3.5M	Run as is	Fine-tuned 1.5M	Scratch 1.5M	Scratch 5.0M
Mean ± stderr	711 ± 59.3	31.9 ± 4.2	$466 \pm 37.9$	399.7 ± 22.5	455.5 ± 33.4	$319.3 \pm 9.7$	$97.5 \pm 17.4$	$11.8 \pm 3.3$	$42.2 \pm 6.4$
% distance > 200	$50.0 \pm 0.0$	0	$64.2 \pm 5.6$	$88.2 \pm 3.3$	$69.6 \pm 5.7$	$50.0 \pm 0.0$	$1.5 \pm 1.4$	0	0
% distance > 400	$35.0 \pm 4.1$	0	$63.6 \pm 6.6$	$33.2 \pm 7.1$	$51.9 \pm 5.7$	$8.4 \pm 2.8$	0	0	0
% distance > 600	35.8 ± 4.5	0	$42.6 \pm 6.1$	$14.9 \pm 4.4$	$28.1 \pm 5.4$	0	0	0	0

Table 1. Quantitative evaluation of the policy learnt on Level-1 of Mario using only curiosity without any reward from the game when run "as is" or when further fine-tuned on subsequent levels. The performance is compared against the Mario agains

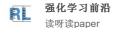
发现在超级马里奥游戏里面,直接把在Level-1使用纯curiosity训练得到的agent放在Level-3里面表现还不错,有319.3分。但是再在Level-3环境里面多训练一些回合之后,分数反而下降了。文中谈到,其原因是Level-3有一个比较难的地方,需要一系列操作组合才能够越过去,纯探索很难做到;当智能体越不过去的时候,智能体的前向模型的预测就越来越准,随之而来的Intrinsic reward就会越来越小,因此得到的结果就会越来越差。

这样越训越差的退化现象是由于reward定义基于一个变化的模型预测来定义的,这可能是未来需要解决的一个问题。

编辑于 2018-10-29



## 文章被以下专栏收录



进入专栏