EXPLORATION BY RANDOM NETWORK DISTILLATION

Yuri Burda* OpenAI
Harrison Edwards* OpenAI
Amos Storkey Univ. of Edinburgh
Oleg Klimov OpenAI

【强化学习算法 26】RND

张楚珩
清华大学 交叉信息院博士在读

RND的全称是random network distillation，是OpenAI在刚刚过去的万圣节挂在arXiv上面的。

原文传送门

Burda, Yuri, et al. "Exploration by Random Network Distillation." arXiv preprint arXiv:1810.12894 (2018).

特色

这还是一篇讲intrinsic reward的文章，目的是为了辅助extrinsic reward，使得智能体更好地探索。本专栏前面讲过好几种intrinsic reward的设计模式了，比如基于动力学模型预测误差的（Curiosity、ICM）；基于各种信息增益的（Empowerment、VIME），这里的方法又是一种新的设计模式。

其设计的初衷主要还是基于对状态访问计数（count-based），但由于是高维连续空间，这个计数更多地可以看做是密度估计。如果类似的状态之前访问地少，那么说明这个状态比较新奇，那么就给予比较高的intrinsic reward。文章用了一个比较机智的方法来快速地估计某个状态是不是之前出现的比较少。

过程

1. 使用神经网络来作为状态新颖程度的估计

我们的目标是找到一种方法对于每一个状态都返回一个值，告诉我们它与之前见到过的状态有多少的相似程度。本文的方法基于一个观察，如果在神经网络训练中，如果有类似的数据出现在训练中，那么其预测误差将会显著减小。

对于状态 x，我们考虑去预测任意一个确定性的函数 $f(x)$，预测的方法就是训练一个神经网络 $\hat{f}(x;\theta)$，在训练样本上面最小化 $\|\hat{f}(x;\theta)-f(x)\|^2$。对于一个新的样本，如果预测误差小，说明这个样本在之前见过（或者见过类似的），那么就给予较低的奖励；反之，说明这个样本之前没怎么见过，于是给予较高的奖励。

个人认为，这种设定下，我们一直保持更新的这个神经网络就相当于在维护的一个density estimator，我们可以随时知道一个新的状态与之前状态分布离得远不远。

2. 神经网络预测误差的构成

文中说到，神经网络预测误差主要由以下四个部分构成

1. 训练数据量和分布：如果类似的数据见得少，那么这种数据的预测误差大；这是我们希望利用的性质，而后面其他的误差来源都是我们不希望见到的，尽可能去避免它们。
2. 目标拟合函数的随机性：预测有误差可能由于希望拟合的函数本身就有随机性；之前很多方法使用拟合的动力学模型来作为目标，然后用动力学模型预测出来的下一步状态的误差作为intrinsic reward。这样的目标随机性就很大，相应的内在奖励说不清楚是由于状态很新颖导致的，还是由于随机性导致的。在这里，就使用了一个确定性的函数 $f(x)$ 来避免这个问题。
3. 模型能力不够：有可能目标拟合函数太过复杂，以至于神经网络根本没可能拟合出来，即最好情况也会产生较大的预测误差。在这里，确定性函数 $f(x)$ 也同样使用一个随机参数的神经网络来生成，避免了这个问题。
4. 学习的过程导致：可能使用的优化方法不够好，导致会产生额外的预测误差。

通过这样的一些分析，我们大致了解了为什么神经网络的预测误差能够作为一个样本是否“眼熟”的衡量依据。文章还引用了另一套理论[1]来说明这里的预测误差等效于模型对于预测这个样本的uncertainty，也就等效于该样本与之前样本分布的“距离”。

3. 与extrinsic reward相结合

文中还提到，对intrinsic reward来说，比较好的方法是做non-episodic的探索，即一个回合结束之后，相应的收益（return）还继续累计而不截断。否则，一旦智能体做出某个动作导致回合结束，那么其收益为0，会激励智能体更加保守而不是更加探索。

但是对于extrinsic reward来说，比较好的方法是做episodic的探索，因为如果把return做累计，智能体可能就不断地在游戏开始的附近刷点小分，然后“自杀”继续回到这个地方刷分。

对于内在奖励和外部奖励这两种不同的探索模式，作者认为同时训练两个价值函数 $V=V_E+V_I$ 。其好处还在于两个价值函数能用不同的discount rate来训练。

算法

Algorithm 1 RND pseudo-code

$N \leftarrow$ number of rollouts
$N_{\text{opt}} \leftarrow$ number of optimization steps
$K \leftarrow$ length of rollout
$M \leftarrow$ number of initial steps for initializing observation normalization
$t = 0$
Sample state $s_0 \sim p_0(s_0)$
for $m = 1$ **to** M **do**
  sample $a_t \sim \text{Uniform}(a_t)$
  sample $s_{t+1} \sim p(s_{t+1}|s_t, a_t)$
  Update observation normalization parameters using s_{t+1}
  t += 1
end for
for $i = 1$ **to** N **do**
  for $j = 1$ **to** K **do**
    sample $a_t \sim \pi(a_t|s_t)$
    sample $s_{t+1}, e_t \sim p(s_{t+1}, e_t|s_t, a_t)$
    calculate intrinsic reward $i_t = \|\hat{f}(s_{t+1}) - f(s_{t+1})\|^2$
    add $s_t, s_{t+1}, a_t, e_t, i_t$ to optimization batch B_i
    Update reward normalization parameters using i_t
    t += 1
  end for
  Normalize the intrinsic rewards contained in B_i
  Calculate returns $R_{I,i}$ and advantages $A_{I,i}$ for intrinsic reward
  Calculate returns $R_{E,i}$ and advantages $A_{E,i}$ for extrinsic reward
  Calculate combined advantages $A_i = A_{I,i} + A_{E,i}$
  Update observation normalization parameters using B_i
  for $j = 1$ **to** N_{opt} **do**
    optimize θ_π wrt PPO loss on batch B_i, R_i, A_i using Adam
    optimize $\theta_{\hat{f}}$ wrt distillation loss on B_i using Adam
  end for
end for

参考文献

[1] Osband, Ian, John Aslanides, and Albin Cassirer. "Randomized Prior Functions for Deep Reinforcement Learning."*arXiv preprint arXiv:1806.03335*(2018).

发布于 2018-11-05

强化学习 (Reinforcement Learning)

文章被以下专栏收录