CTOR-MIMIC
EEP MULTITASK AND TRANSFER REINFORCEMENT
EARNING

nilio Parisotto, Jimmy Ba, Ruslan Salakhutdinov
partment of Computer Science
iversity of Toronto
ronto, Ontario, Canada
parisotto,jimmy,rsalakhu}@cs.toronto.edu

【强化学习 85】Actor-mimic

张楚珩
清华大学 交叉信息院博士在读

算法名字就叫做 Actor-mimic，主要用途为 transfer learning。

原文传送门

Parisotto, Emilio, Jimmy Lei Ba, and Ruslan Salakhutdinov. "Actor-mimic: Deep multitask and transfer reinforcement learning." arXiv preprint arXiv:1511.06342 (2015).

特色

Actor-mimic 主要解决的是如何是把之前在多任务上面学习到的知识（比如之前训练得到的一些 policy network 或者 value network）迁移到一个单一的策略神经网络上。

过程

1. 问题设定

假设有 N 个 source games $s_1,\cdots,s_N$ ，每个游戏都有一个训练好的 DQN 网络， $E_1,\cdots,E_N$ 。现在需要训练一个策略网络，使得该策略网络能够在每一个游戏中都有比较好的表现。

一种比较直接的方式，就是把这些已经学习到的 DQN 网络用作 target 新训练一个 DQN 去预测不同 action 的 Q 值。但是由于不同游戏之间 reward 的 scale 和分布都不一样，因此这种方法可能有一些问题。

文中使用的方法是先把这些 DQN 网络转化为一个相应的 Boltzmann policy，然后最小化这些 policy 和所需要得到的策略网络之间的差距。

2. Actor-mimic

Actor-mimic 的目标是训练一个策略网络，这里称作 multitask Actor-Mimic Network（AMN）。

一个训练好的 DQN 对应的 Boltzmann policy 为：

$$\pi_{E_i}(a|s) = \frac{e^{\tau^{-1}Q_{E_i}(s,a)}}{\sum_{a' \in \mathcal{A}_{E_i}} e^{\tau^{-1}Q_{E_i}(s,a')}}, \tag{3}$$

where τ is a temperature parameter and $\mathcal{A}_{E_i}$ is the action space used by the expert E_i, $\mathcal{A}_{E_i} \subseteq \mathcal{A}$.

训练的目标是最小化上述策略和 AMN 之间的 crossentropy：

$$\mathcal{L}^i_{policy}(\theta) = \sum_{a \in \mathcal{A}_{E_i}} \pi_{E_i}(a|s) \log \pi_{\text{AMN}}(a|s;\theta), \tag{4}$$

值得注意的是，原则上来说，如果 source DQN 都给定了，如果需要 uniformly 地去把 DQN 里面的信息迁移出来，其实可以在状态空间和行动空间均匀采样然后去优化上述 loss 即可，即，不需要再和环境交互（sample complexity = 0）。但是显然，事实上即使训练好的 DQN 也只是在真实样本分布上有效，因此这里还是需要对于环境中的状态进行采样，并在这些样本上来学习。因此这里的 sample complexity 唯一的目的就是为了得到状态样本的分布，这样想来好像有点『亏』？

在这篇文章的实验中，他们每次使用最新的 AMN+epsilon-greedy 来采集样本。

3. Feature regression objective

由于已知各个 source DQN，而把 DQN 从中间某一层切开，其实也可以分为 feature extraction 和 Q network 的部分，假设这两部分都是可以得到的。同样地，AMN 也可以看做 feature extraction 和 policy network 的部分。设某个专家网络的 feature extraction 部分为 $h_{E_i}(s)$ 。这里希望能够利用这一部分信息来指导 AMN 中 feature extraction 部分的学习。

文中使用如下损失函数

$$\mathcal{L}^i_{FeatureRegression}(\theta, \theta_{f_i}) = \|f_i(h_{\text{AMN}}(s;\theta);\theta_{f_i}) - h_{E_i}(s)\|_2^2, \tag{5}$$

即要求，AMN 中的 feature extraction 部分通过某个函数 $f_i(\cdot;\theta_{f_i})$ 的映射之后和相应 source DQN 的 feature 一致。通过这种方式，AMN 的 feature extraction 能够包含所有 source DQN hidden 层的信息。

加上前面的 policy loss，Actor-mimic 的总体训练损失函数如下：

$$\mathcal{L}^i_{ActorMimic}(\theta, \theta_{f_i}) = \mathcal{L}^i_{policy}(\theta) + \beta * \mathcal{L}^i_{FeatureRegression}(\theta, \theta_{f_i}), \tag{6}$$

4. Convergence properties

分析过程做了两个简化：1）feature extraction 部分去掉了；2）只分析单个 source DQN 的情况。证明部分感觉很绕，不是很看得懂。。。

Actor-mimic 学习的过程为可以写为

$$\min_{\theta} \mathop{\mathbb{E}}_{s \sim D^{\pi_{\text{AMN}}, \epsilon\text{-greedy}}(\cdot)} \left[\mathcal{H}\Big(\pi_E(a|s),\ \pi_{\text{AMN}}(a|s;\theta) \Big) \right] + \lambda \|\theta\|_2^2, \tag{7}$$

注意到在和 AMN 相关的项既出现在期望里面，也出现在期望的分布上，这对于分析造成了困难。为了证明通过梯度下降方法，优化该目标能够得到一个还不错的结果，因此文章把对于它的分析分为了三个部分。先证明期望分布部分不变，通过对于上述目标做梯度下降，能够收敛到一个稳定的点；再证明分布不变收敛到稳定的点之后，更新新一轮的 π_{AMN} ，这样迭代下去能够收敛到一个稳

定的点；最后利用了一个 online learning 的结论直接说明了学习到的 π_{AMN} 相比于 π_E 损失的差距。

第一步中，认为期望下面的分布是固定的，然后对上述目标做梯度下降，梯度下降每一轮中参数的变化可以写为

$$\Delta\theta_t = -\alpha_t \left[\Phi^T D_\pi (P_{\theta_{t-1}} - \Pi_E) + \lambda \theta_{t-1} \right]. \tag{8}$$

这里假设了线性拟合，即 π_{AMN} 是由一个拟合的 Q 函数生成的 $p(a|s,\theta) \propto \exp \hat{Q}(s,a;\theta)$ ，而这个 Q 函数是线性拟合的， $\hat{Q}(s,a;\theta) = \phi(s)^T \theta_a$ 。上述的 Φ, P 都是相应的矩阵形式， D_π 是状态概率分布， Π_E 是 source DQN 策略的矩阵表示。

这样能够证明，在合适的条件下，做梯度下降能够收敛到一个固定的点（相对于 θ 导数为零）。

Lemma 1. *Under a fixed policy π^* and a learning rate schedule that satisfies $\sum_{t=1}^{\infty} \alpha_t = \infty$, $\sum_{t=1}^{\infty} \alpha_t^2 < \infty$, the parameters θ, updated by the stochastic gradient descent learning algorithm described above, asymptotically almost surely converge to a unique solution θ^*.*

第二步中假设第一步收敛之后，再重新采样（期望下面的分布），然后再重复第一步，文章证明说这样也能收敛到一个固定的策略。

Theorem 1. *Assume the Markov decision process is irreducible and aperiodic for any policy π induced by the Γ operator and Γ is Lipschitz continuous with a constant c_ϵ, then the sequence of policies and model parameters generated by the iterative algorithm above converges almost surely to a unique solution π^* and θ^*.*

想要证明这件事情，只需说明每更新一步，都是一个 contraction。假设 softmax 操作 $\Gamma(Q)$ 产生一个相对于这个价值函数的 Boltzmann policy。证明可以顺着

$$||\Gamma(\hat{Q}_{\theta_1}) - \Gamma(\hat{Q}_{\theta_2})|| \le a||\hat{Q}_{\theta_1} - \hat{Q}_{\theta_2}|| \le b||\theta_1 - \theta_2|| \le c||\pi^1 - \pi^2||$$

其中最后一步是和更新的具体过程有关。

第三步说明最后得到的结果性能不太差。

Proposition 1. *For the iterative algorithm described in Section (4.2), if the loss function in Eq. (7) converges to ϵ with the solution π_{AMN} and $Z^{\pi^*}_{T-t+1}(s,\pi^*) - Z^{\pi^*}_{T-t+1}(s,a) \ge u$ for all actions $a \in \mathcal{A}$ and $t \in \{1, \cdots, T\}$, then the cost-to-go of Actor-Mimic $J_T(\pi_{AMN})$ grows linearly after executing T actions: $J_T(\pi_{AMN}) \le J_T(\pi_E) + uT\epsilon/\log 2$.*

好像套用了文章引用的一个结论，完全不明白。

实验

文章做了两个实验：一个是 multi-task learning，即在不同任务上学习 expert DQN 的同时，也去把信息提取到 AMN 中；另一个是 transfer learning，即把 expert DQN 中的信息提取到 AMN 之后，把其权重作为初始化来继续分别在各个任务上训练。

我感觉我不是很能理解到文章给定的两个任务的做法和意义。如果按照我这样理解，multi-task learning 里面为什么非要把学习到的 expert DQN 里面的知识再转移一下？训练好了各自的 expert DQN 之后对于不同的任务直接用不好？对于 transfer learning，要想它比较有意义可能就是在一组任务上训练好之后，把知识都放到一个 AMN 中，然后使用这个 AMN 作为初始化学习，学习地更快（更节省样本）。但是这里面可能是 positive transfer 也可能是 negative transfer。我理解 transfer learning 的终极目标还是对于每个任务平均来讲需要的采样数目降下来，即 sample efficiency，但是目前都基本上只能说在特定相似的任务们之间有 positive transfer。

如果有大佬看到，可以评论解释一下 multi-task learning 的意义，感谢！以及 transfer learning 目前也没有比较有好的结果，比如能保证 positive transfer 之类的。

发布于 2019-08-04

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