Importance Weighted Evolution Strategies

Víctor Campos
Barcelona Supercomputing Center
victor.campos@bsc.es

Xavier Giro-i-Nieto
Universitat Politècnica de Catalunya
xavier.giro@upc.edu

Jordi Torres
Barcelona Supercomputing Center
jordi.torres@bsc.es

【强化学习 43】IW-ES

张楚珩
清华大学 交叉信息院博士在读

全称为Importance Weighted Evolution Strategies（正如标题）。

原文传送门

Campos, Víctor, Xavier Giro-i-Nieto, and Jordi Torres. "Importance Weighted Evolution Strategies." arXiv preprint arXiv:1811.04624 (2018).

特色

本文之前讲过一篇使用演化算法来优化强化学习问题中神经网络的文章（【强化学习算法9】ES，ES 文章），这里算是一个对于前者工作的改进，改进的主要目的是保持较好的可并行性的基础上，增加算法的 sample efficiency。

过程

1. 背景

强化学习利用使用 ES 的最大**优势**在于它能够以很小的通讯开销（communication overhead） 做并行，以节省 wall-clock time。即在每个 worker 上面运行相互独立的 rollouts，得到最后的总奖励之后，把它返回给 master，由 master 汇总并决定这一步的更新。在 ES 文章里面使用了 common random number 的技术，使得 master 和 worker 之间只需要通讯一个标量就可以得到全部神经网络参数的信息，这减小了通讯开销。

ES 方法最大的**缺点**在于效率十分低下，需要很多样本（多次与环境交互）才能够学习到好的策略，这是因为它几乎没有使用到强化学习中 temporal difference、boostrap、off-policy等加快学习的技术。本文为了提高 ES 的效率，提出每采样一批样本，都重复利用几轮这批样本做更新。

2. 方法

原本的 ES 的方法就是在神经网络的参数空间上做扰动，并且把扰动后的神经网络作为策略去运行 rollouts，把 rollouts 获得的总奖励作为 fittness function 来做更新。经过简单的推导，可以得到神经网络参数的更新公式

$$\nabla_\theta F(\theta) \approx \frac{1}{n\sigma^2} \sum_{i=1}^{n} F(\theta + \epsilon_i)\epsilon_i \tag{2}$$

其中 $F(\cdot)$ 表示 fitness function的数值。

这里考虑每一批样本 $\{\epsilon_i, F(\theta, \epsilon_i)\}$ 得到之后，利用这一批样本做多轮更新。标记 $\theta^t \in \mathbb{R}^{|\theta|}$ 为第 t 轮更新之后的神经网络参数， $\epsilon_i^t \in \mathbb{R}^{|\theta|}$ 为之前采样得到的扰动。在这种情况下，神经网络的更新公式可以写为

$$\nabla_\theta F(\theta) \approx \frac{1}{\sigma^2 \sum_i c_i} \sum_{i=1}^{n} F(\theta^t + \epsilon_i^t)(\theta^t + \epsilon_i^t - \theta^{t+k})c_i$$

注意到 $\theta^t + \epsilon_i^t$ 就是原采样时使用的神经网络参数，它与 θ^{t+k} 相减就表示当前神经网络参数要位移多少才能到达 $\theta^t + \epsilon_i^t$ ，这和前面公式中的单个 ϵ_i 是类似的。

注意到这里各个样本不再是等权重的了，而是各自附加了一个权重，这个权重的含义是原来采样的那个样本离目前最新的这个神经网络参数是否仍然足够近，如果太远可能之前的信息就不够有效地来指征目前这个参数附近的梯度了。这个权重的计算公式如下

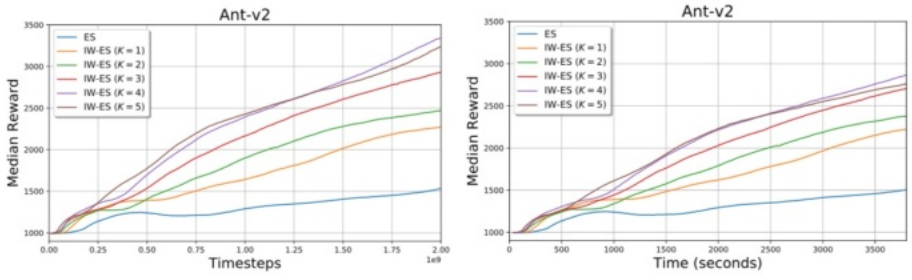
$$c_i = \frac{N(\theta^t + \epsilon_i^t - \theta^{t+k}; 0, \sigma^2\mathbf{I})}{N(\epsilon_i^t; 0, \sigma^2\mathbf{I})} = \frac{\prod_{j=1}^{|\theta|} N(\theta_j^t + \epsilon_{i,j}^t - \theta_j^{t+k}; 0, \sigma^2)}{\prod_{j=1}^{|\theta|} N(\epsilon_{i,j}^t; 0, \sigma^2)}$$

3. 并行性

这个权重的计算中，分母的计算是可以并行的，但是分子上的计算只能放到 master 节点上计算，这是由于公式里面含有当前这一轮更新的最新参数，这个参数还没传到 worker 上。

实验结果

这篇文章只做了 Mujoco Ant-v2 的实验，每批样本更新多轮能够提升 sample efficiency。



(a) (b)

Figure 1: Performance of ES and IW-ES as a function of **(a)** the number of interactions with the environment, and **(b)** wall-clock time. K denotes the number of additional importance weighted updates after each standard update. We observe that additional updates increase the data efficiency of the method in the low learning rate regime, but performing too many importance weighted updates can be detrimental due to an increased variance, e.g. $K = 5$ underperforms $K = 4$. A similar trend is observed in terms of wall-clock time.

同时，当模型参数较多的时候，计算耗时会增加，这是由于 master 上串行计算权重造成的。不过，本文把 ES 用到了最多含有324K个参数的神经网络上，这算是一个在较大规模神经网络上用ES的比较成功的尝试。

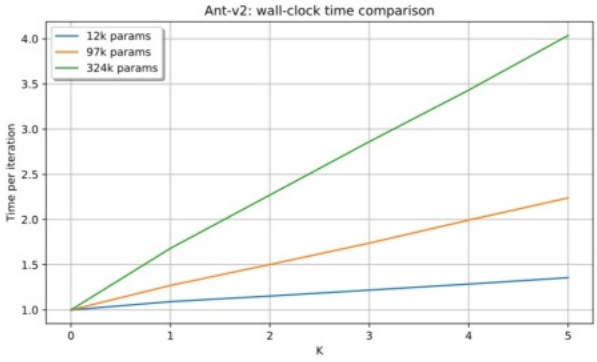


Figure 3: Time per iteration for different values of K, normalized by the time taken by ES (i.e. $K = 0$). Our implementation parallelizes the computation of importance weights only across the CPU cores in the node hosting the master process, which becomes a bottleneck for larger models.

另外，该算法在learning rate较大的时候容易翻车，这是由于learning rate较大，这样之前采样得到的样本在更新若干轮之后可利用的价值就不大了，这样权重就会有很大的variance，算法更不稳定，容易翻车。

补充

文章总计了一些提高 sample efficiency 的方法：使用 model-based，使用 off-policy，on-policy 中每批样本多用几轮。

发布于 2019-03-03

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