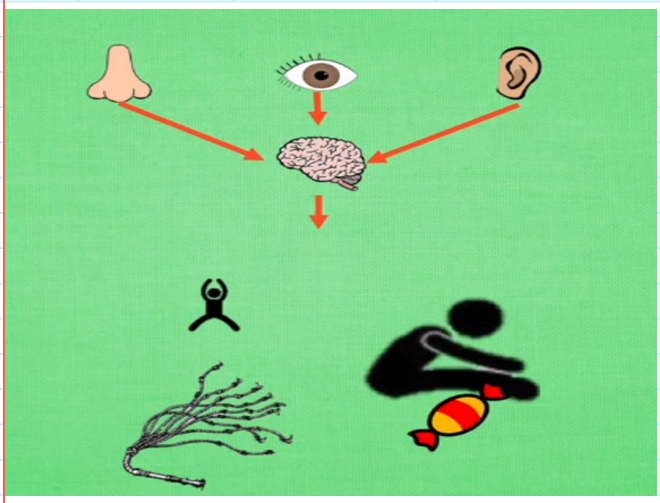
Policy Gradients

2019年7月26日 15:36

一、算法思想

On policy

可以采用神经网络，但没有误差函数来反向传递



1）算法思想：

输入的特征使神经网络最终输出选中一个的动作，反向传播使其下次能够再次选中同一个行为的概率增大，reward的好坏会来影响反向传递的情况，使得下次选到该行为的概率增大或减小。

2）特点：

Policy Gradients不是Value-based 方法 (Q learning, Sarsa), 但他也要接受环境信息 (observation), 不同的是他要输出不是 action 的 value, 而是具体的那一个 action, 这样 policy gradient 就跳过了 value 这个阶段

算法输出的是**动作的概率**，而不是Q值。

损失函数的形式为：**loss= -log(prob)*vt**

需要一次**完整的episode**才可以进行参数的更新

二、更新算法

回合更新

基于整条回合数据的更新， 是基于连续情况

动作选择：不再是更具Q值来选取动作，而是根据概率进行选择，所以不用再随机化，且不再采用贪心策略。

来源：https://blog.csdn.net/qq_36829091/article/details/83213707

function REINFORCE

Initialise θ arbitrarily

for each episode $\{s_1, a_1, r_2, \ldots, s_{T-1}, a_{T-1}, r_T\} \sim \pi_\theta$ **do**

for $t = 1$ to $T - 1$ **do**

$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t) v_t$

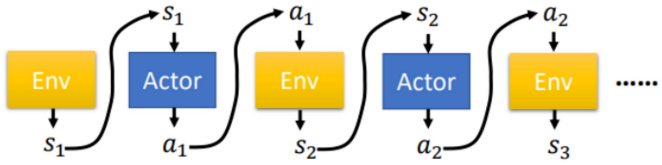
end for

end for

return θ

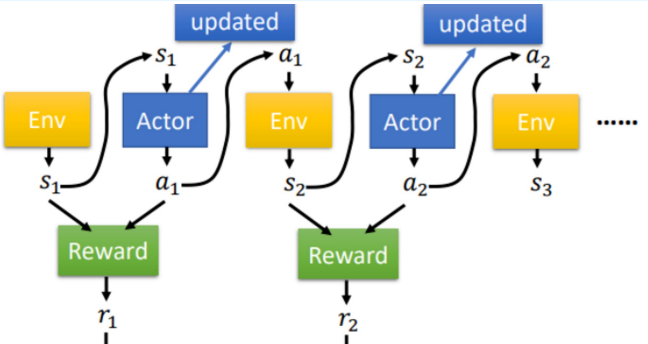
end function

Vt表示在当前状态s下选择相应a的奖励



Trajectory $\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$

$$p_\theta(\tau)$$
$$= p(s_1)p_\theta(a_1|s_1)p(s_2|s_1, a_1)p_\theta(a_2|s_2)p(s_3|s_2, a_2)\cdots$$
$$= p(s_1)\prod_{t=1}^{T} p_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)$$



Expected Reward

$$\bar{R}_\theta = \sum_\tau R(\tau)p_\theta(\tau) = E_{\tau\sim p_\theta(\tau)}[R(\tau)] \qquad R(\tau) = \sum_{t=1}^{T} r_t$$

计算使得R最大化

Policy Gradient $\bar{R}_\theta = \sum_\tau R(\tau)p_\theta(\tau)$ $\nabla\bar{R}_\theta = ?$

$$\nabla\bar{R}_\theta = \sum_\tau R(\tau)\nabla p_\theta(\tau) = \sum_\tau R(\tau)p_\theta(\tau)\frac{\nabla p_\theta(\tau)}{p_\theta(\tau)}$$

$R(\tau)$ do not have to be differentiable

It can even be a black box.

$$= \sum_\tau R(\tau)p_\theta(\tau)\nabla log p_\theta(\tau)$$

$\nabla f(x) = f(x)\nabla log f(x)$

$$= E_{\tau\sim p_\theta(\tau)}[R(\tau)\nabla log p_\theta(\tau)] \approx \frac{1}{N}\sum_{n=1}^{N} R(\tau^n)\nabla log p_\theta(\tau^n)$$

$$= \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla log p_\theta(a_t^n|s_t^n)$$

反向梯度的计算方式：

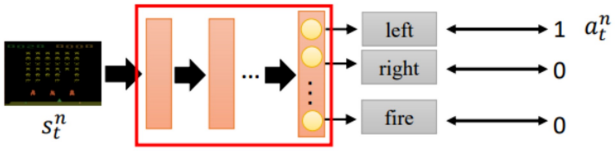
Implementation

$\theta \leftarrow \theta + \eta\nabla\bar{R}_\theta$

$$\nabla\bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla log p_\theta(a_t^n|s_t^n)$$

Consider as classification problem

s_t^n a_t^n $R(\tau^n)$



$$\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} log p_\theta(a_t^n|s_t^n) \Rightarrow \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\nabla log p_\theta(a_t^n|s_t^n)$$

TF, pyTorch ...

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n) log p_\theta(a_t^n|s_t^n) \Rightarrow \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} R(\tau^n)\nabla log p_\theta(a_t^n|s_t^n)$$

小技巧一：

Tip 1: Add a Baseline

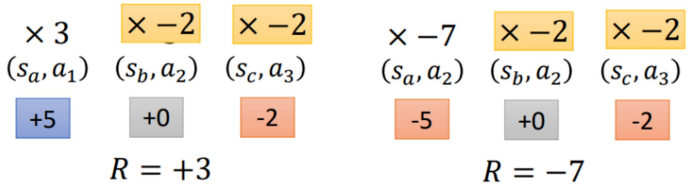
$\theta \leftarrow \theta + \eta\nabla\bar{R}_\theta$ It is possible that $R(\tau^n)$ is always positive.

$$\nabla\bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(R(\tau^n) - b)\nabla log p_\theta(a_t^n|s_t^n) \qquad b \approx E[R(\tau)]$$

因为我们不可能穷举完所以的情况，只是不断的取sample而已，有些情况可能没有被sample到，如果reward总是正的的话，会导致该情况更难被sample到。通过减去一个baseline使得Reward有正有负。

小技巧二：

Tip 2: Assign Suitable Credit



$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}(R(\tau^n) - b)\nabla log p_\theta(a_t^n|s_t^n)$$

$$\sum_{t'=t}^{T_n} r_{t'}^n$$

Reward改成选择该action后的所有reward之和，因为在次之前的r与该action无关。

进阶：可以给reward附加一个衰减值（不断衰减）

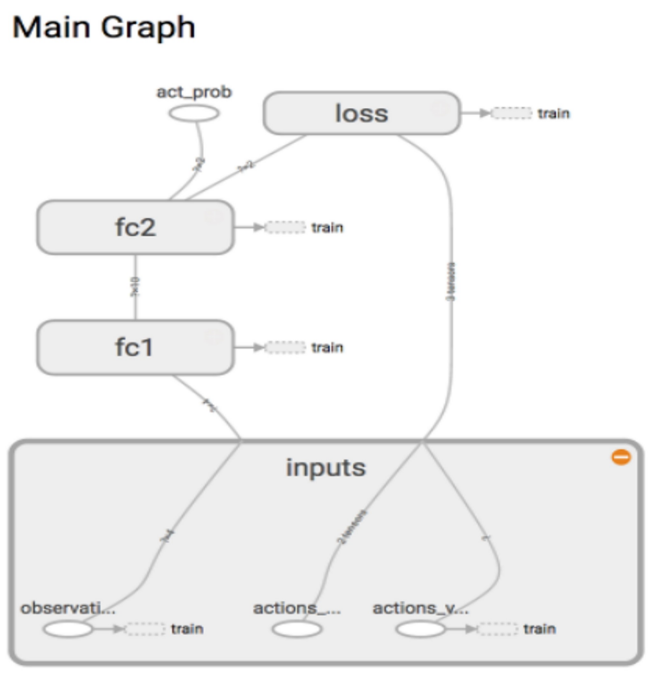
$$\sum_{t'=t}^{T_n} r_{t'}^n \rightarrow \sum_{t'=t}^{T_n} \gamma^{t'-t} r_{t'}^n$$

Add discount factor $\gamma < 1$

so**最终形式**为：

$$\nabla \bar{R}_\theta \approx \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n}\left(\sum_{t'=t}^{T_n}\gamma^{t'-t} r_{t'}^n - b\right)\nabla log p_\theta(a_t^n|s_t^n)$$

三、网络结构



目标函数：带衰减的reward的累加和期望。

$L(\theta) = \mathbb{E}(r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots | \pi(,\theta))$

使之最大

损失函数：

$L(\theta) = \sum log\pi(a|s,\theta) f(s,a)$

一个动作所获得的reward多就增大其概率，反之则反

log项表示输出的概率，f(s,a)表示该动作的好与坏

四、代码分析

https://github.com/957001934/Reinforcement-Learning

神经网络的输入为：input: observations

labels:actions

output:probability

损失函数：Loss=cross entropy(layer out,labels)
输出的R=probability*action_value
通过最大化动作的价值来调整参数，使得最大价值的动作出现概率更高
reward使用了decay，the oldest data,the decay less