TTIC 31230, Fundamentals of Deep Learning

David McAllester, Winter 2020

The Policy as a Q-Function

Analysis

Simulations select $\text{argmax}_a\ U(s,a)$.

$$U(s,a) = \begin{cases} \lambda_u\ \pi_\Phi(s,a) & \text{if } N(s,a) = 0 \\ \\ \hat{\mu}(s,a) + \lambda_u\ \pi_\Phi(s,a)/N(s,a) & \text{otherwise} \end{cases} \tag{1}$$

$$\Phi^* \ = \ \underset{\Phi}{\text{argmin}}\ E_{(s,\pi,R)\sim\text{Replay},\ a\sim\pi} \left(\begin{array}{l} (V_\Phi(s) - R)^2 \\ \\ -\lambda_\pi \log \pi_\Phi(a|s) \\ \\ +\lambda_R\ ||\Phi||^2 \end{array} \right) \tag{2}$$

Equation (2) establishes the meaning of $\pi_\Phi(a|s)$ as a stochastic policy.

Analysis

$$U(s,a) = \begin{cases} \lambda_u \, \pi_\Phi(s,a) & \text{if } N(s,a) = 0 \\ \\ \hat{\mu}(s,a) + \lambda_u \, \pi_\Phi(s,a)/N(s,a) & \text{otherwise} \end{cases} \tag{1}$$

But equation (1) then seems ill-typed — how can we add a reward and a probability?

The types would work if we use $Q_\Phi(s,a)$ rather than $\pi_\Phi(s,a)$.

Analysis

$$\Phi^* = \underset{\Phi}{\operatorname{argmin}} \; E_{(s,\pi,R)\sim\text{Replay},\; a\sim\pi} \left(\begin{array}{l} (V_\Phi(s) - R)^2 \\ \\ -\lambda_\pi \log \pi_\Phi(a|s) \\ \\ +\lambda_R \, ||\Phi||^2 \end{array}\right) \quad (2)$$

It is not clear why this use of a policy as a Q-funtion is so effective.

One explanation might be that a policy is discriminative — it is trained on its ability to discriminate between actions rather than its ability to assign a value to each action. Q-value to the actions.

Analysis

$$\Phi^* \;=\; \underset{\Phi}{\mathrm{argmin}} \;\; E_{(s,\pi,R)\sim\mathrm{Replay},\; a\sim\pi} \left(\begin{array}{l} (V_\Phi(s) - R)^2 \\ \\ -\lambda_\pi \log \pi_\Phi(a|s) \\ \\ +\lambda_R \, ||\Phi||^2 \end{array} \right) \qquad (2)$$

This works in tree search bootstrapping but it is not clear whether one can replace the Q-function critic with a policy in a general actor-critic algorithm.

END