CS 6101 Week #4 Notes: Actor-Critic Introduction, Value Functions and Q-Learning

Note taking: Alexandre Gravier, Joel Lee
LaTeX transcription: Alexandre Gravier <al.gravier@gmail.com>

Contents

1 Recap about policy gradients

We define $J(\theta) \doteq E_{\tau \sim p_\theta(\tau)}\left[\sum_t r\left(\mathbf{s}_t, \mathbf{a}_t\right)\right]$ so that the objective of RL can be defined as an optimization exercise consting in finding an assignment of policy parameters $\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} J(\theta)$.

$J(\theta)$ is not usually optimizable as such due to f.e. dimensionality issues, so we use a sample-based unbiased estimate: $J(\theta) \approx \frac{1}{N}\sum_i \sum_t r\left(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}\right)$. Taking the gradient of $J(\theta)$ along θ allows maximizing the expected reward as per the policy.

1.1 Policy differentiation with a "convenient identity"

Let $r(\tau) \doteq \sum_{t=1}^{T} r\left(\mathbf{s}_t, \mathbf{a}_t\right)$ the total reward of a trajectory τ.

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta E_{\tau \sim p_\theta(\tau)}[r(\tau)] && && \text{(1a)}\\
&= \nabla_\theta \int \pi_\theta(\tau) r(\tau) d\tau && \text{by definition of expectation} && \text{(1b)}\\
&= \int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau && \text{by linearity} && \text{(1c)}
\end{aligned}$$

At this point, the expression $\int \nabla_\theta \pi_\theta(\tau) r(\tau) d\tau$ seems rather intractable. This is where the following "convenient identity" can be used to derive a tractable expression of $\nabla_\theta J(\theta)$.

A convenient identiy

$$\pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau) = \pi_\theta(\tau)\frac{\nabla_\theta \pi_\theta(\tau)}{\pi_\theta(\tau)} = \nabla_\theta \pi_\theta(\tau) \tag{2}$$

Furthermore, we can expand the definition of $\pi_\theta(\tau)$ and take its logarithm:

$$\pi_\theta(\tau) = p(\mathbf{s}_1) \prod_{t=1}^{T} \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\, p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \tag{3a}$$

$$\Leftrightarrow \log \pi_\theta(\tau) = \log p(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t) \tag{3b}$$

Using (2) and (3b) in (1c), the gradient of the objective becomes:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int \pi_\theta(\tau) \nabla_\theta \log \pi_\theta(\tau) r(\tau) d\tau && \text{using (2)} && (4a)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \log \pi_\theta(\tau) r(\tau)\right] && \text{by definition of expectation} && (4b)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \left[\log p(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\right] r(\tau)\right] && && (4c)
\end{aligned}$$

We note that in the expression $\nabla_\theta \left[\log p(\mathbf{s}_1) + \sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) + \log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)\right]$ of the gradient w.r.t. θ in (4c), the terms $\log p(\mathbf{s}_1)$ and $\log p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ are independent of θ, so we are left with:

$$\begin{aligned}
\nabla_\theta J(\theta) &= E_{\tau \sim p_\theta(\tau)} \left[\nabla_\theta \left[\sum_{t=1}^{T} \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right] r(\tau)\right] && && (5a)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right) r(\tau)\right] && \text{by linearity} && (5b)\\
&= E_{\tau \sim p_\theta(\tau)} \left[\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right) \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right] && \text{by definition of } r(\tau) && (5c)
\end{aligned}$$

In Equation (5c), the gradient of J is now a computable function of π_θ only.

We earlier mentioned the sample estimate of $J(\theta) \approx \frac{1}{N} \sum_i \sum_t r(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$; similarly $\nabla_\theta J(\theta)$ is approximated with samples, leading us to the algorithm:

$$\textbf{REINFORCE algorithm:} \begin{cases} \text{sample } \{\tau^i\} \text{ from } \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t) \\ \nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\right) \sum_{t=1}^{T} r(\mathbf{s}_t, \mathbf{a}_t)\right) \\ \theta \leftarrow \theta + \alpha \nabla_\theta J(\theta) \end{cases} \tag{6}$$

In (6), there is no use of the Markov property, so the algorithm can be used as such on POMDPs.

1.2 The bad news

We introduce some simplifying notation:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_t \mid \mathbf{s}_t \right) \right) \sum_{t=1}^{T} r\left(\mathbf{s}_t, \mathbf{a}_t \right) \right) \tag{7a}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \nabla_\theta \log \pi_\theta \left(\tau \right) r\left(\tau \right) \tag{7b}$$

The big problem with vanilla REINFORCE lies in variance: if you were to repeatedly collect a small finite number N of samples, estimating each time the gradient based on these samples, you would observe that these estimates of the gradient vary a lot.

That means that the gradient descent step will likely take us away from the goal, and convergence will be very slow, or may not happen.

Additionally, given two sets of sample trajectories differing only by a constant factor, but dofferently centred around a total reward of 0, the basic policy gradient algorithm in (6) may change the parameters very differently.

There are two tricks that help with the large varaiance of the samples, b oth of which should always be used because they have no drawbacks.

1.3 Variance reduction with causality: the "rewards to go" trick

In (6), the gradient approximation can be improved by making use of causality: the policy at time t' cannot affect the reward at time t when $t < t'$.

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_t \mid \mathbf{s}_t \right) \right) \sum_{t=1}^{T} r\left(\mathbf{s}_t, \mathbf{a}_t \right) \right) \tag{8a}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \left(\underbrace{\nabla_\theta \log \pi_\theta \left(\mathbf{a}_t \mid \mathbf{s}_t \right)}_{\textit{gradient at } \mathrm{t}} \underbrace{\sum_{t'=1}^{T} r\left(\mathbf{s}_{t'}, \mathbf{a}_{t'} \right)}_{\textit{this total reward should be computed from time } \mathrm{t}} \right) \right) \quad \text{by distributivity} \tag{8b}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \left(\nabla_\theta \log \pi_\theta \left(\mathbf{a}_t \mid \mathbf{s}_t \right) \underbrace{\sum_{t'=t}^{T} r\left(\mathbf{s}_{t'}, \mathbf{a}_{t'} \right)}_{\textit{reward to go}} \right) \right) \quad \text{by causality} \tag{8c}$$

Intuitively, the "**rewards to go**" trick comes from the observation that at time t, all past rewards cannot be affected by policy decisions.

We define the "reward to go" function $\hat{Q} : [\![1, N]\!] \times [\![1, T]\!] \mapsto \mathbb{R}$ as:

$$\hat{Q}_{i,t} \doteq \sum_{t'=t}^{T} r\left(\mathbf{s}_{t'}, \mathbf{a}_{t'} \right) \tag{9}$$

Therefore, the gradient of the objective function approximation with rewards to go is:

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_t \mid \mathbf{s}_t \right) \hat{Q}_{i,t} \tag{10}$$

References