Policy Gradient Derivations

Aaron Lou

May 2018

1 Formulation

We formulate the policy θ as follows

$$p_\theta(s_1, a_1, \ldots, s_T, a_T) = \pi_\theta(\tau) = p(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)p(s_{t+1}|s_t a_t)$$

For the finite and infinite horizon cases, the value we want to maximize θ^* are (respectively)

$$\arg\max_\theta \sum_{t=1}^{T} E_{(s_t,a_t)\sim p_\theta(s_t,a_t)}[r(s_t,a_t)] \quad \arg\max_\theta E_{(s,a)\sim p_\theta(s,a)}[r(s,a)]$$

2 Derivation

We start by defining

$$J(\theta) = E_{\tau\sim\pi_\theta(\tau)}\sum_t r(s_t,a_t)$$

We need to find $\nabla_\theta J(\theta)$, ie the gradient we wish to use. We notice

$$J(\theta) = \int \pi_\theta(\tau)r(\tau)d\tau \implies \nabla_\theta J(\theta) = \int \nabla_\theta \pi_\theta(\tau)r(\tau)d\tau$$

We notice that

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

Therefore, it follows that

$$\nabla_\theta J(\theta) = \int \pi_\theta(\tau)\nabla_\theta \log(\pi_\theta(\tau))r(\tau)d\tau = E_{\tau\sim p_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau)r(\tau)]$$

We can calculate this by noting that

$$\pi_\theta(\tau) = p(s_1)\prod_{t=1}^{T}\pi_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)$$

$$\implies \log \pi_\theta(\tau) = \log p(s_1) + \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t) + \log p(s_{t+1}|s_t, a_t)$$

$$\implies \nabla_\theta \log \pi_\theta(\tau) = \sum_{t=1}^{T} \log \pi_\theta(a_t|s_t)$$

Notice that this value only depends on our current policy. Using Monte Carlo, we approximate $J(\theta)$ to be

$$J(\theta) \approx \frac{1}{N}\sum_i \sum_t r(s_{i,t}, a_{i,t}) \implies \nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}(\sum_{t=1}^{T} \log \pi_\theta(a_t|s_t))(\sum_{t=1}^{T} r(s_{i,t}, a_{i,t}))$$

We can finish by using gradient ascent on our policy θ

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

for learning rate α.

3 Improvements

Problem: high variance problem. This occurs because shifting the reward by a constant creates massively different changes in gradients.

Solution:

1) Causality: policy at time t' can't affect $t < t'$.

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}(\sum_{t'=1}^{T} \log \pi_\theta(a_{t'}|s_{t'}))(\sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'}))$$

We often write $\hat{Q}_{i,t} = \sum_{t'=t}^{T} r(s_{i,t'}, a_{i,t'})$

2) Baseline: add a baseline to the expected values.

$$\nabla_\theta J(\theta) \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \log \pi_\theta(\tau)[r(\tau) - b]$$

$$b = \frac{1}{N}\sum_{i=1}^{N} r(\tau)$$

Derivation: want to show that adding this to our policy gradient won't change the bias

$$E[\nabla_\theta \log \pi_\theta(\tau)b] = \int \pi_\theta(\tau)\nabla_\theta \log \pi_\theta(\tau)bd\tau = \int \nabla_\theta \pi_\theta(\tau)bd\tau$$
$$= b\nabla_\theta \int \pi_\theta(\tau)d\tau = b\nabla_\theta 1 = 0$$

Derivation of best baseline (not average): calculate the variance to be

$$\text{Var} = E_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \pi_\theta(\tau)(r(\tau)-b)]^2 - E_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \pi_\theta(\tau)(r(\tau)-b)]^2$$
$$\implies \frac{d\text{Var}}{db} = \frac{d}{db}E[g(\tau)^2(r(\tau)-b)^2] = \frac{d}{db}(-2E[g(\tau)^2 r(\tau)b] + b^2 E[g(\tau)^2])$$
$$= -2E[g(\tau)^2 r(\tau)] + 2bE[g(\tau)^2] = 0$$
$$\implies b = \frac{E[g(\tau)^2 r(\tau)]}{E[g(\tau)^2]}$$

4 In Tensorflow

Implement a pseudo-loss (maximum likelihood) weighted by $\hat{Q}$. Use softmax cross entropy with logits and multiply by q values.