Policy Gradient [2]

DOROZHKO Anton

Novosibirsk State University

June 6, 2020

[2] SpinningUP RL research : Part 3

Outline

Q-learning

Action-Value approximation

- Find w that minimizes loss between $Q^\pi(s,a)$ and $\hat{Q}(s,a;w)$
- MSE

$$J(w) = \mathbb{E}_\pi[(Q^\pi(s,a) - \hat{Q}(s,a;w))^2]$$

- SGD samples the gradient

$$-\frac{1}{2}\nabla_w J(w) = \mathbb{E}_\pi[(Q^\pi(s,a) - \hat{Q}(s,a;w))\nabla_w \hat{Q}^\pi(s,a;w)]$$

$$\Delta w = -\frac{1}{2}\alpha \nabla_w J(w)$$

Novosibirsk State University
*THE REAL SCIENCE

Incremental Model-Free Control

- MC

$$\Delta w = \alpha(G_t - \hat{Q}^{\pi}(s_t, a_t; w))\nabla_w \hat{Q}(s_t, a_t; w)$$

- SARSA

$$\Delta w = \alpha(r + \gamma \hat{Q}(s', a'; w) - \hat{Q}^{\pi}(s_t, a_t; w))\nabla_w \hat{Q}(s_t, a_t; w)$$

- Q-learning

$$\Delta w = \alpha(r + \gamma \max_{a'} \hat{Q}(s', a'; w) - \hat{Q}^{\pi}(s_t, a_t; w))\nabla_w \hat{Q}(s_t, a_t; w)$$

Novosibirsk State University
*THE REAL SCIENCE

What if ?

What if we can directly optimize our expected reward w.r.t. the parameters of our policy ?

Policy Optimization

- Stochastic, parametrized policy π_θ
- maximize expected return

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[G_t(\tau)]$$

- SGD with **policy gradient**

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\pi_\theta)|_{\theta_k}$$

Novosibirsk State University
*THE REAL SCIENCE

How to compute policy gradient

- Derive the analitycal gradient
- Compute sample estimate

1) Probability of a Trajectory

Trajectory

$$\tau = (s_0, a_0, ..., s_{T+1})$$

Probability of a Trajectory

$$P(\tau|\theta) = \rho_0(s_0) \prod_{t=0}^{T} P(s_{t+1}|s_t, a_t)\pi_\theta(a_t, s_t)$$

Take log

$$logP(\tau|\theta) =$$

Novosibirsk State University
*THE REAL SCIENCE

2) The Log-Derivative Trick

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

Rearrange

$$\nabla_\theta P(\tau|\theta) = P(\tau|\theta) \nabla_\theta log P(\tau|\theta)$$

Novosibirsk State University
*THE REAL SCIENCE

3) Gradients of Environment Functions

- $\rho_0(s_0)$, $P(s_{t+1}|s_t, a_t)$ and $G_t(\tau)$
- no dependence on θ
- Gradients w.r.t θ are **zero**

Novosibirsk State University
*THE REAL SCIENCE

Grad-Log-Prob of Trajectory

$$\nabla_\theta \log P(\tau|\theta) = \cancel{\nabla_\theta \log \rho_0(s_0)} + \sum_{t=0}^{T} \Big(\cancel{\nabla_\theta \log P(s_{t+1}|s_t, a_t)} + \nabla_\theta \log \pi_\theta(a_t|s_t) \Big)$$

$$= \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t).$$

$\nabla_\theta J(\pi_\theta) = \dots$

Basic Policy Gradient

Derivation for Basic Policy Gradient

$$\begin{aligned}
\nabla_\theta J(\pi_\theta) &= \nabla_\theta \underset{\tau\sim\pi_\theta}{\mathrm{E}}[R(\tau)] & \\
&= \nabla_\theta \int_\tau P(\tau|\theta)R(\tau) & \text{Expand expectation} \\
&= \int_\tau \nabla_\theta P(\tau|\theta)R(\tau) & \text{Bring gradient under integral} \\
&= \int_\tau P(\tau|\theta)\nabla_\theta \log P(\tau|\theta)R(\tau) & \text{Log-derivative trick} \\
&= \underset{\tau\sim\pi_\theta}{\mathrm{E}}[\nabla_\theta \log P(\tau|\theta)R(\tau)] & \text{Return to expectation form} \\
\therefore \nabla_\theta J(\pi_\theta) &= \underset{\tau\sim\pi_\theta}{\mathrm{E}}\left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)R(\tau)\right] & \text{Expression for grad-log-prob}
\end{aligned}$$

How to compute expectation ?

$$\mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) R(\tau)]$$

$$\hat{g} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) R(\tau)$$

Novosibirsk State University
*THE REAL SCIENCE

Basic Policy Gradient in words

- Define policy as parametrized function
- Write the gradient of the loss function
- Play games collect trajectories (1 or more)
- Compute cumulative discounted rewards for each t
- Compute gradients
- Update parameters of your policy
- **Comment: throw out trajectories**

Loss is not a normal loss

1. Data is dependent on the parameters
2. Is not connected to performance $J_(\pi_\theta)$ after one step
3. So, minimization of this "loss" itself for several steps on the same batch will not give better rewards.

Consider only future rewards

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t)R(\tau)]$$

Consider only future rewards

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) R(\tau)]$$

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) \sum_{t'=t}^{T} R(s_{t'}, a_{t'}, s_{t'+1})]$$

[20]SpinningUP RL research : Part 3

Lemma

$$\mathbb{E}_{x \sim P_\theta}[\nabla_\theta log P_\theta(x)] = 0$$

Proof: distributions are normalized $\rightarrow$ take gradient $\rightarrow$ log derivative trick

$$\int_x P_\theta(x) = 1$$

$$\nabla_\theta \int_x P_\theta(x) = \nabla_\theta 1 = 0$$

$$\nabla_\theta \int_x P_\theta(x) = \int_x P_\theta(x) \nabla_\theta log P_\theta(x) = \mathbb{E}_{x \sim P_\theta}[\nabla_\theta log P_\theta(x)]$$

Proof of go-to-reward usage

$$\nabla_\theta J_\theta(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) \sum_{t'=t}^{T} R(s_{t'}, a_{t'}, s_{t'+1})]$$

Consider only future rewards

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) \sum_{t'=t}^{T} R(s_{t'}, a_{t'}, s_{t'+1})]$$

Why it's better ?
Recall the sum of random variables with 0 mean and some variance.

Improvement: Baseline

Consequence of lemma

For $\forall b(s)$:

$$\mathbb{E}_{a_t \sim \pi_\theta}[\nabla_\theta log \pi_\theta(a_t|s_t) b(s_t)] = 0$$

This allows:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^{T} \nabla_\theta log \pi_\theta(a_t|s_t) \left(\sum_{t'=t}^{T} R(s_{t'}, a_{t'}, s_{t'+1}) - b(s_t) \right)]$$

We can choose b to reduce variance. In practice $b(s_t) = V^\pi(s_t)$ value function works well.

Basic Policy Gradient

Algorithm 1 Vanilla Policy Gradient Algorithm

Input: initial policy parameters θ_0, initial value function parameters ϕ_0

for $k = 0, 1, 2, ...$ **do**

  Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.

  Compute rewards-to-go $\hat{R}_t$.

  Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.

  Estimate policy gradient as

$$\hat{g}_k = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)|_{\theta_k} \hat{A}_t.$$

  Compute policy update, either using standard gradient ascent,

$$\theta_{k+1} = \theta_k + \alpha_k \hat{g}_k,$$

  or via another gradient ascent algorithm like Adam.

  Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

  typically via some gradient descent algorithm.

end for

Policy Gradient with automatic differentiation

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} \nabla_\theta log \pi_\theta(a_{i,t}|s_{i,t}) \hat{G}_{i,t}$$

How can we compute policy gradient with autodiff ?
Create a function s.t. it's gradient is policy gradient.
You will implement "pseudo-loss"

$$J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{T} log \pi_\theta(a_{i,t}|s_{i,t}) \hat{G}_{i,t}$$

cross entropy (discrete) or squared error (gaussian)

It is good if you can ...

1. Define Markov Decision Process
2. Describe some task in terms of MDP
3. Understand value and action-value functions
4. Be able to apply Policy Iteration and Value Iteration
5. Understand model-based vs model-free
6. Be able to apply MC, TD(0) algorithms to Q-function
7. Be able to apply Q-learning, SARSA, Expected value SARSA
8. Describe the "Exploitation / Exploration" dilemma
9. Understand ϵ-Greedy and "Optimism in the face of uncertainty" (UCB) ideas
10. Understand the idea of Policy Optimization (Policy Gradient)