Part 6: Policy gradients

EL 2805 – Reinforcement Learning

Alexandre Proutiere

KTH, The Royal Institute of Technology

Objectives of this part

- An example: learning to play the rock-scissors-paper game
- Stochastic gradient descent algorithm
- Generic policy gradient algorithm
- REINFORCE algorithm: Monte Carlo policy gradient

Outline

1. The rock-scissors-paper game
2. Stochastic gradient descent algorithm
3. Policy gradients in episodic RL problems
4. Policy gradients in discounted RL problems

References

- Barto-Sutton's book chapter 13
- Stochastic gradient lecture notes:
  `https://stanford.edu/~jduchi/PCMIConvex/Duchi16.pdf`
- REINFORCE algorithm: R.J. Williams, Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning (1992)

Part 6: Outline

1. Rock-scissors-paper game
2. Stochastic gradient descent algorithm
3. Finite time horizon RL problems
    a. The policy gradient theorem
    b. REINFORCE algorithm
    c. Variance reduction techniques
4. Infinite horizon discounted RL problems

1. The rock-scissors-paper game

You and your opponent simultaneously select R, S, or P. R wins over S, S wins over P, and P wins over R.
In each round, your opponent selects R, S, P in an i.i.d. manner according to an **unknown** distribution $\mu = (\mu_R, \mu_S, \mu_P)$. How can you sequentially select your action to learn the optimal policy?

MDP model

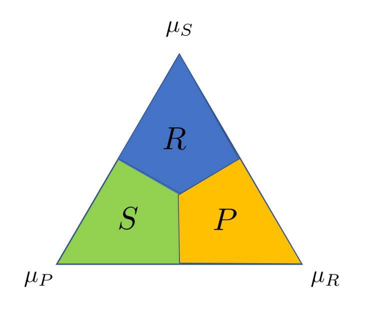
- Time horizon: $T = 1$
- No state, no transition probabilities
- Set of actions: $\{R, S, P\}$
- Rewards: given the selected action a, the reward is random variable $r(a)$ such that:
  - If $a = R$, $r(a) = 1$ with probability μ_S and $r(a) = 0$ with probability $1 - \mu_S$
  - If $a = S$, $r(a) = 1$ with probability μ_P and $r(a) = 0$ with probability $1 - \mu_P$
  - If $a = P$, $r(a) = 1$ with probability μ_R and $r(a) = 0$ with probability $1 - \mu_R$
- Randomized policies: a policy is defined by a distribution $\theta = (\theta_R, \theta_S, \theta_P)$ over actions ($\theta_R + \theta_S + \theta_P = 1$)
- Objective: find a policy θ maximizing $\mathbb{E}_\theta[r(a)] = \mathbb{E}_{a\sim\theta}[r(a)]$

Randomized policies and their return

- Return under the policy θ:

$$J(\theta) = \mathbb{E}_{a\sim\theta}[r(a)] = (\theta_R \mu_S + \theta_S \mu_P + \theta_P \mu_R)$$

- Optimal policy:



Learning the optimal policy

- The random strategy μ of the opponent is unknown
- How can we learn the optimal policy by sequentially interacting with the opponent?
- A solution: the Stochastic Gradient Descent algorithm

- **Projected Gradient Descent (PGD) algorithm:**
  Let $\mathcal{C}$ be a convex subset of $\mathbb{R}^n$, and let $f : \mathcal{C} \to \mathbb{R}$ be a convex function.
  Objective: find $x^\star \in \mathcal{C}$ such that $f(x^\star) \leq f(x)$ for all $x \in \mathcal{C}$.

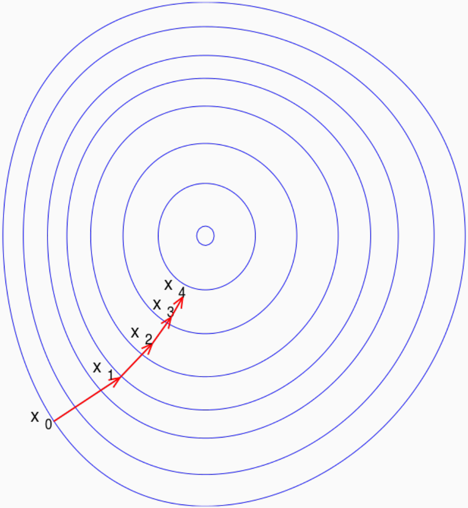
PGD Algorithm:

1. **Initialization:** $x^{(0)} \in \mathcal{C}$
2. **Iterations:** for $k \geq 0$,

$$x^{(k+1)} = P_{\mathcal{C}}(x^{(k)} - \alpha_k \nabla f(x^{(k)}))$$

   where $P_{\mathcal{C}}$ is the projection on $\mathcal{C}$: $P_{\mathcal{C}}(y) = \inf_{x \in \mathcal{C}} \|y - x\|_2$.

Projected gradient descent algorithm



Convergence of the PGD algorithm

Let $f_{\min}^{(k)} = \min_{i=0,\dots,k} f(x^{(i)})$.

Theorem. Assume that for all x, $\|\nabla f(x)\|_2 \le G$.
We have for all $k \ge 1$:

$$f_{\min}^{(k)} - f(x^\star) \le \frac{\|x^{(0)} - x^\star\|_2^2 + G^2 \sum_{i=0}^k \alpha_i^2}{2 \sum_{i=0}^k \alpha_i}$$

- Constant step sizes: $\alpha_i = \alpha$. $f_{\min}^{(k)}$ converges within $G^2\alpha/2$ of the optimal as $k \to \infty$.
- Square summable but not summable step sizes: $\sum_{i=0}^\infty \alpha_i = \infty$, $\sum_{i=0}^\infty \alpha_i^2 < \infty$. $f_{\min}^{(k)}$ converges to the optimal value as $k \to \infty$.

Proof

Convexity. for all x, y, $f(y) \geq f(x) + \nabla f(x) \cdot (y - x)$.
Projection. By projection we get closer to all points inside $\mathcal{C}$:
$\|P_{\mathcal{C}}(z) - x\|_2 \leq \|z - x\|_2$ for all $x \in \mathcal{C}$, and all z

$$\begin{aligned}
\|x^{(k+1)} - x^\star\|_2^2 &\leq \|x^{(k)} - \alpha_k \nabla f(x^{(k)}) - x^\star\|_2^2 \\
&= \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k \nabla f(x^{(k)}) \cdot (x^{(k)} - x^\star) + \alpha_k^2 \|\nabla f(x^{(k)})\|_2^2 \\
&\leq \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k (f(x^{(k)}) - f(x^\star)) + \alpha_k^2 G^2
\end{aligned}$$

Applying the above recursively, we get:

$$\|x^{(k+1)} - x^\star\|_2^2 \leq \|x^{(0)} - x^\star\|_2^2 - 2\sum_{i=0}^{k} \alpha_i (f(x^{(i)}) - f(x^\star)) + \sum_{i=0}^{k} \alpha_i^2 G^2$$

Hence: $f_{\min}^{(k)} - f(x^\star) \leq \frac{\|x^{(0)} - x^\star\|_2^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2\sum_{i=0}^{k} \alpha_i}$

Example: Rock-Scissors-Paper game

Cost of the policy $\theta = (\theta_R, \theta_S, 1 - \theta_R - \theta_S)$:

$$f(\theta) = -J(\theta) = -\left(\theta_R(\mu_S - \mu_R) + \theta_S(\mu_P - \mu_R) + \mu_R\right)$$

It is linear and hence convex.

$$\nabla f(\theta) = \begin{pmatrix} (\mu_R - \mu_S) \\ (\mu_R - \mu_P) \end{pmatrix}$$

We do not know μ, so the gradient is unknown! We cannot apply the PGD algorithm.

However we have an unbiased estimator: at the k-th round of the game, for $a \in \{R, S, P\}$

$$\mu_a = \mathbb{E}\left[\frac{\text{nb\_times\_opp\_played a}}{k}\right]$$

The estimator can be used in a Stochastic PDG algorithm ...

Stochastic Gradient Descent algorithm

- Let $f : \mathcal{C} \to \mathbb{R}$ be a convex function.
- Unbiased estimator of the gradient: $g(x)$ is a r.v. such that $\nabla f(x) = \mathbb{E}[g(x)]$

SGD Algorithm:

1. **Initialization:** $x^{(0)}$
2. **Iterations:** for $k \geq 0$,

$$x^{(k+1)} = x^{(k)} - \alpha_k g(x^{(k)})$$

Convergence of the SGD algorithm

Let $f_{\min}^{(k)} = \min_{i=0,\dots,k} f(x^{(i)})$.

Theorem. Assume that for all x, $\mathbb{E}[\|\nabla f(x)\|_2^2] \le G^2$.
We have for all $k \ge 1$:

$$\mathbb{E}[f_{\min}^{(k)} - f(x^\star)] \le \frac{\|x^{(0)} - x^\star\|_2^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2 \sum_{i=0}^{k} \alpha_i}$$

- Same convergence results as those for the non-stochastic GD algorithm but in expectation.

Proof

Using the same arguments (in expectation) as before, we get:

$$\mathbb{E}[\|x^{(k+1)} - x^\star\|_2^2] \leq \|x^{(0)} - x^\star\|_2^2 - 2\sum_{i=0}^{k} \alpha_i \mathbb{E}[f(x^{(i)}) - f(x^\star)] + \sum_{i=0}^{k} \alpha_i^2 G^2$$

Hence: $\min_{i=0,\ldots,k} \mathbb{E}[f(x^{(i)} - f(x^\star)] \leq \frac{\|x^{(0)} - x^\star\|_2^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2\sum_{i=0}^{k} \alpha_i}$

Jensen inequality: For any convex function $f : \mathbb{R}^n \to \mathbb{R}$, for any r.v. X, $\mathbb{E}[f(X)] \geq f(\mathbb{E}[X])$.

We conclude that:

$$\mathbb{E}[f_{\min}^{(k)} - f(x^\star)] \leq \min_{i=0,\ldots,k} \mathbb{E}[f(x^{(i)} - f(x^\star)] \leq \frac{\|x^{(0)} - x^\star\|_2^2 + G^2 \sum_{i=0}^{k} \alpha_i^2}{2\sum_{i=0}^{k} \alpha_i}$$

Example: Rock-Scissors-Paper game

Cost of the policy $\theta = (\theta_R, \theta_S, 1 - \theta_R - \theta_S)$:

$$f(\theta) = -J(\theta) = -\left(\theta_R(\mu_S - \mu_R) + \theta_S(\mu_P - \mu_R) + \mu_R\right)$$

$$\nabla f(\theta) = \begin{pmatrix} (\mu_R - \mu_S) \\ (\mu_R - \mu_P) \end{pmatrix}$$

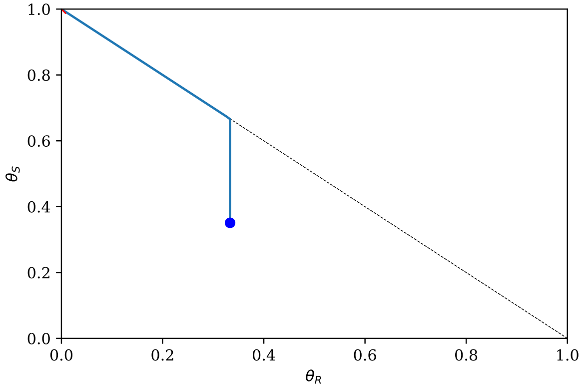
Unbiased gradient estimators: For $a \in \{R, S, P\}$,
$\hat{\mu}_a = \frac{\text{nb\_times\_opp\_played a}}{k}$ (empirical estimator)
or $\hat{\mu}_a = 1_{\{\text{opp.plays a at time k}\}}$ (instantaneous estimator)

$$g(\theta) = \begin{pmatrix} (\hat{\mu}_R - \hat{\mu}_S) \\ (\hat{\mu}_R - \hat{\mu}_P) \end{pmatrix}$$

Projected SGD algorithm: $\theta^{(k+1)} = P_{\mathcal{C}}\left(\theta^{(k)} - \alpha_k g(\theta^{(k)})\right).$

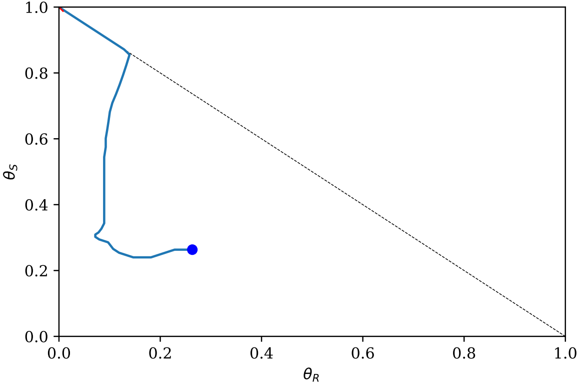
Example: Rock-Scissors-Paper game

True gradient available: PGD algorithm, step sizes = 0.07



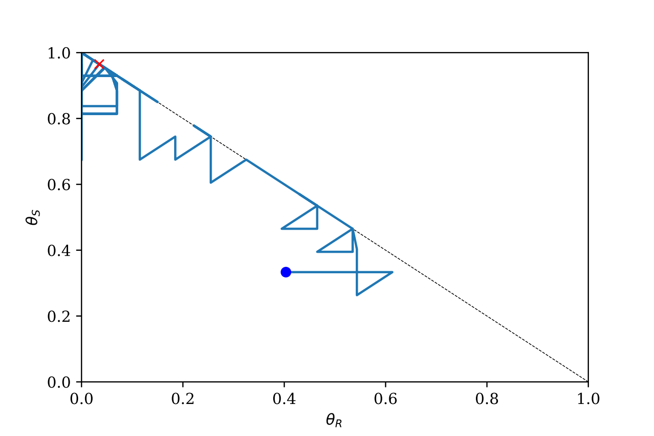
Example: Rock-Scissors-Paper game

Empirical gradient estimator: SGD algorithm, step sizes = 0.07



Example: Rock-Scissors-Paper game

Instantaneous gradient estimator: SGD algorithm, step sizes = 0.07



3. Finite time horizon RL problems

- Finite time horizon MDP:
  - Time horizon T
  - State space: S
  - Actions available in state $s \in S$, A_s ($A = \cup_{s \in S} A_s$)
  - Initial state $s_1 \sim p(\cdot)$
  - Stationary transition probabilities $p(s'|s, a)$ and deterministic rewards $r(a, s)$
  - Objective: for an initial state s, find a policy π maximizing (over all possible policies)

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{T} r(s_t^\pi, a_t^\pi) | s_1^\pi = s\right] = \mathbb{E}_\pi\left[\sum_{t=1}^{T} r(s_t, a_t) | s_1 = s\right]$$

Parametrized randomized policies

- Parameter $\theta \in \mathbb{R}^n$, policy π_θ defined by $\pi_\theta(s,a)$ the probability to take action a in state s
- Example: softmax policies. Encode a preference in taking action a in state s as $h(s,a,\theta)$ and define

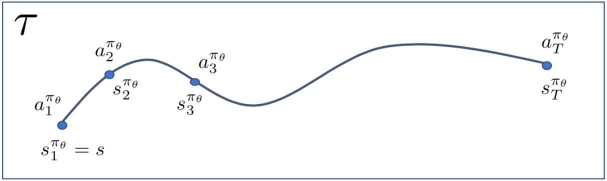
$$\pi_\theta(s,a) = \frac{e^{h(s,a,\theta)}}{\sum_b e^{h(s,b,\theta)}}$$

- Main advantage: compared to deterministic policy, smoothly varying policies w.r.t. θ. The optimal policy can be learnt using gradient descent as shown below.
- Value function of π_θ:

$$V^{\pi_\theta}(s) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} r(s_t,a_t) | s_1 = s\right]$$

Objective function and its gradient

Objective: maximize $J(\theta) = \mathbb{E}_{s_1 \sim p}[V^{\pi_\theta}(s_1)]$
Trajectory: $\tau = (s_1, a_1, \ldots, s_T, a_T)$



Total reward of τ: $R(\tau) = \sum_{t=1}^{T} r(s_t, a_t)$
Probability of observing τ under π_θ:

$$\pi_\theta(\tau) = p(s_1)\pi_\theta(s_1, a_1)p(s_2|s_1, a_1)\ldots p(s_T|s_{T-1}, a_{T-1})\pi_\theta(s_T, a_T)$$

Computing the gradient of $J(\theta)$

$$J(\theta) = \sum_{\tau} \pi_\theta(\tau) R(\tau)$$

Hence:

$$\begin{aligned}
\nabla J(\theta) &= \sum_{\tau} (\nabla \pi_\theta(\tau)) R(\tau) \\
&= \sum_{\tau} \pi_\theta(\tau) \times (\nabla \log \pi_\theta(\tau) R(\tau)) \\
&= \mathbb{E}_{\pi_\theta} [\nabla \log \pi_\theta(\tau) R(\tau)]
\end{aligned}$$

with

$$\nabla \log \pi_\theta(\tau) = \sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)$$

The policy gradient theorem

Theorem. *The gradient of the objective function w.r.t. the policy is:*

$$\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=1}^{T} r(s_t, a_t)\right)\right]$$

$\nabla_\theta \pi_\theta(s, a)$ is referred to as the *score function*

REINFORCE algorithm

If we generate an episode following π_θ: $(s_1 = s, a_1, r_1 \dots s_T, a_T, r_T)$ where $r_t = r(s_t, a_t)$ is the observed reward, the quantity $\left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=1}^{T} r_t\right)$ is an unbiased estimator of $\nabla J(\theta)$.

REINFORCE Algorithm:

1. **Initialization:** select $\theta^{(0)}$ arbitrarily
2. **Iterations:** For all $k \geq 0$, for episode k, generate a trajectory under $\pi_{\theta^{(k)}}$: $(s_{1,k} = s, a_{1,k}, r_{1,k}, \dots s_{T,k}, a_{T,k}, r_{T,k})$
   Update the parameter

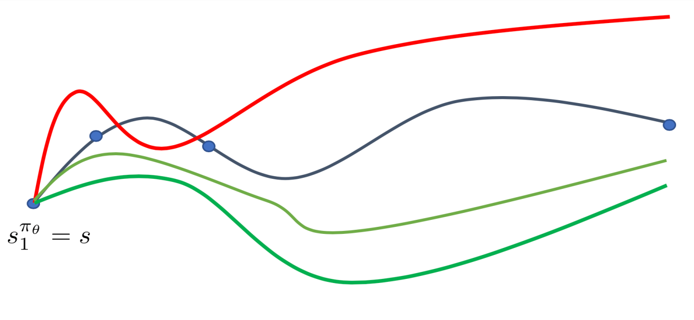
$$\theta^{(k+1)} = \theta^{(k)} + \alpha_k \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_{t,k}, a_{t,k})\right)\left(\sum_{t=1}^{T} r_{t,k}\right)$$

Gradient variance reduction # 1

Let $X = \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right) \left(\sum_{t=1}^{T} r_t\right)$ be the estimated gradient observing one trajectory under π_θ.

Instead, generate n trajectories in an i.i.d. manner under π_θ before updating the policy: $X_1, \dots, X_n$ and compute the estimated gradient as $\frac{1}{n}\sum_{i=1}^{n} X_i$. Then

$$\mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n}\mathrm{Var}(X)$$



Gradient variance reduction # 2

Exercice 1. (See Exercise session) Let $u < t$. Then we have:

$$\mathbb{E}_{\pi_\theta}[\nabla \log \pi_\theta(s_t, a_t) r_u] = 0.$$

We deduce that:

$$\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t) \underbrace{\sum_{u=t}^{T} r(s_u, a_u)}_{R_t\text{:reward to go}}\right]$$

Update with reduced variance:

$$\theta \leftarrow \theta + \alpha \sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t) \sum_{u=t}^{T} r(s_u, a_u)$$

Gradient variance reduction # 3

Baseline. Adding a baseline helps.

$$X = \underbrace{\left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right)}_{\nabla\text{(log-likelihood of the traj.)}} \underbrace{\left(\sum_{t=1}^{T} r_t\right)}_{\text{return}}$$

If returns are always positive, the likelihood of the corresponding trajectories are increased ... and it will take a long time before the most desirable actions are favoured.

Solution: center the returns on a baseline b.

$$X = \left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right) \left(\sum_{t=1}^{T} r_t - b\right)$$

Gradient variance reduction # 3

With a baseline, the gradient estimator remains unbiased:

$$\mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(\tau) b] = 0.$$

Natural baseline: b is the empirical mean of the return of the n generated episodes (if n epsiodes are used per policy update):

$$b = \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} r(s_{t,i}, a_{t,i})$$

Variance-optimal constant baseline: b minimizing Var(X) is

$$b = \frac{\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2 R(\tau)]}{\mathbb{E}_{\pi_\theta}[(\nabla_\theta \log \pi_\theta(\tau))^2]}$$

Gradient variance reduction # 3

Time-dependent baseline: when n episodes are used per policy update, and the reward-to-go is used,

$$b_t = \frac{1}{n} \sum_{i=1}^{n} \sum_{u=t}^{T} r(s_{u,i}, a_{u,i})$$

Update:

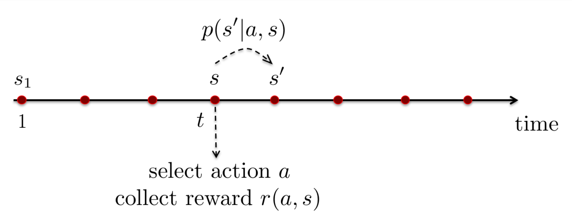
$$\theta \leftarrow \theta + \alpha \frac{1}{n} \sum_{i=1}^{n} \sum_{t=1}^{T} \nabla \log \pi_\theta(s_{t,i}, a_{t,i}) (\sum_{u=t}^{T} r_{u,i} - b_u)$$

Examples using Deep Learning

Pong game: `https://www.youtube.com/watch?v=YOW8m2YGtRg`

TRPO + GAE (Mujoco simulator)
`https://sites.google.com/site/gaepapersupp/`

4. Infinite horizon discounted RL problems



Discounted RL problems:

- **Unknown** stationary transition probabilities $p(s'|s,a)$ and rewards $r(s,a)$, uniformly bounded: $\forall a,s,\ |r(s,a)| \le 1$
- Objective: for a given discount factor $\lambda \in [0,1)$, from the data, find a policy $\pi^\star \in MD$ maximizing (over all possible policies)

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{T} \lambda^{t-1} r(s_t^\pi, a_t^\pi,)|s_1^\pi = s\right]$$

Objective function and its gradient

Objective: maximize $J(\theta) = \mathbb{E}_{s_1 \sim p}[V^{\pi_\theta}(s_1)]$
Discounted stationary distribution ρ_θ under π_θ:

$$\forall s \in \mathcal{S}, \quad \rho_\theta(s) = (1-\lambda) \sum_{s'} p(s') \sum_{k=0}^{\infty} \lambda^k \mathbb{P}_{\pi_\theta}[s_k = s | s_1 = s']$$

Theorem. *The gradient of the objective function w.r.t. the policy is:*

$$\nabla J(\theta) = \frac{1}{1-\lambda} \mathbb{E}_{s \sim \rho_\theta, a \sim \pi_\theta(s,\cdot)} \left[\nabla \log \pi_\theta(s,a) Q^{\pi_\theta}(s,a) \right]$$

We need a critic to estimate Q^{π_θ} ... see the lecture on actor-critic algorithms.

Summary

Policy gradient algorithms assume that:

- Policies are parametrized: $\pi \in \{\pi_\theta : \theta \in \Theta\}$
- Maximize $J(\theta) = \mathbb{E}_{s_1 \sim p}[V^{\pi_\theta}(s_1)]$
- $J(\theta)$ must be smooth in θ, and preferably concave!
- PG algorithms are SGD algorithms to find a maximizer of J

Policy gradient theorems:

- Episodic RL:
  $\nabla J(\theta) = \mathbb{E}_{\pi_\theta}\left[\left(\sum_{t=1}^{T} \nabla \log \pi_\theta(s_t, a_t)\right)\left(\sum_{t=1}^{T} r(s_t, a_t)\right)\right]$
- ∞-horizon RL:
  $\nabla J(\theta) = \frac{1}{1-\lambda}\mathbb{E}_{s\sim\rho_\theta, a\sim\pi_\theta(s,\cdot)}\left[\nabla \log \pi_\theta(s,a) Q^{\pi_\theta}(s,a)\right]$