RL-Week-8

L1 Policy Gradient Algos.

- Policy search → instead of maintaining estimates of value func^{n}, search in the space of polices.
  - ↳ simple, better convergence, robust to partial observability.
  - ↳ Direct policy search → eg. genetic algos. (more popular)
  - ↳ we will use policy gradient approaches.
  - ↳ most of the times description of policy π is much simpler than the value func^{n}. Better to go for better approximator than going for wrong final values.
- What is partial observability? → We don't have access to true state. But we get something like contextual state. Then, we add more such contextual states, to estimate the further values.
- Policy Gradient Methods
  - ↳ depends on θ → (action preferences, mean, variance, weights of a neural network.
  - → The idea is to modify the current policy params directly instead of

estimating the acⁿ values.

Max. $J(\theta) = E(r_t)$ → performance obj. funcⁿ for policy π.

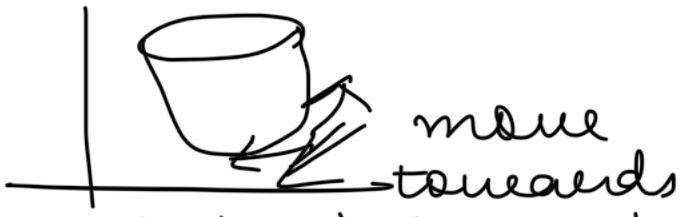
θ Updaⁿ rule $\Rightarrow$ $\theta \leftarrow \theta + \alpha \nabla J(\theta)$

$J(\theta) = E(r_t) = E(G_t)$

↳ expected total reward.

take as small step in the direcⁿ of the gradient

GD



steepest ascent, move towards it.

- let's talk about simplified setting
  ↳ immediate reward or multi-arm bandit.

set of n-arms, pick the arm with highest expected reward.

$$J(\theta) = E(r_t) = \sum_a q_*(a)\, \pi_\theta(a)$$

$q_*(a)$: true expected reward ↓ mean of the gaussian.

- Stochastic Gradient Ascent

$$\nabla J(\theta) = \sum_a q_*(a) \nabla \pi_\theta(a) = \mathbb{E}_{\pi_\theta}[q_* \nabla \ln \pi_\theta(a)]$$

$$= \sum_a q_*(a) \frac{\nabla \pi_\theta(a)}{\pi_\theta(a)} \pi_\theta(a)$$

I don't know q_*. ⇒ sample it.

Now estimate the gradient from N samples.

$$\hat{\nabla} J(\theta) = \frac{1}{N} \sum_{i=1}^{N} u_i \frac{\nabla \pi_\theta(a_i)}{\pi_\theta(a_i)}$$

↙ kind of importance sampling.

c/d the likelihood ratio.

- Reinforce → step sizes are small enough, eventually you will converge to local maxima.

Incrementally

$$\Delta\theta_t = \alpha u_t \frac{\nabla \pi_\theta(a_t)}{\pi_\theta(a_t)}$$

$$\Delta\theta_t = \alpha u_t \frac{\partial \ln \pi_\theta(a_t)}{\partial \theta}$$

With baseline → it helps to stabilize the learning

$$\Delta\theta_t = \alpha (u_t - b_t) \frac{\partial \ln \pi_\theta(a_t)}{\partial \theta}$$

reinforcement baseline ↙ ↘ charac. eligibility

b_t should not depend on a_t

α_t s " " " r_t ⇒ otherwise it leads to bias.

L2 Reinforce

↳ Williams → 1992 → with arbitary rewards.

Eg. binary bandit problem → generalised L_{R-I}

$$\pi(\theta, a) = \begin{cases} \theta & \text{if } a=1 \\ 1-\theta & a=0 \end{cases} \qquad \frac{\partial \ln \pi}{\partial \theta} = \frac{a-\theta}{\theta(1-\theta)}$$

baseline ← $b=0$ & $\alpha = \beta \cdot \theta(1-\theta)$

$$\boxed{\Delta\theta = \beta \cdot r \cdot (a-\theta)}$$

$a=1$	$r>0$	$\theta\uparrow$	$r<0$	$\theta\downarrow$
$a=0$	$r>0$	$\theta\downarrow$	$r>0$	$\theta\uparrow$

$r=0$ ⇒ no change

if penalty, no Δ, if +ve reward, we react to it.

Eg. Softmax

set a baseline to get avg. of obs. rewards.

$$b_t = \bar{r}_t = \bar{r}_{t-1} + \beta \cdot (r_t - \bar{r}_{t-1})$$

we modify the softmax action selection.

$$\Delta\theta_i = \alpha \cdot (r - \bar{r})(1 - \pi(\theta, a_i)) \rightarrow b^t$$

$$\pi(\theta, a_i) = \frac{e^{\theta_i}}{\sum_{j=1}^{n} e^{\theta_j}}$$

$$\frac{\partial \ln \pi(\theta, a_i)}{\partial \theta_i} = 1 - \pi(\theta, a_i)$$

this is how we compute the characteristic eligibility for softmax action selection.

Eg. For continuous actions
Gaussian distribution to select action

$$\pi(a, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(a-\mu)^2}{2\sigma^2}}$$

$$\Delta\mu = \alpha \cdot (r - \bar{r})(a - \mu)$$

$$\Delta\sigma = (\alpha/\sigma) \cdot (r - \bar{r})((a-\mu)^2 - \sigma^2)$$

$$\alpha_\mu = \alpha_\sigma = \alpha\sigma^2$$

$$b_\mu = b_\sigma = \bar{r}$$

L3 Policy Gradient Theorem

- for episodic task, we define the performan

by assuming that every episode starts from state s_0 (non-random)

$$J(\theta) = v_{\pi_\theta}(s_0)$$

↳ true value func^n given a policy π_θ

- Th. (Marbach.)

$$\nabla \bar{v}(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a) \nabla \pi(a|s,\theta)$$

It has a analytical expression for the performance gradient with respect to policy param θ.

$$\nabla v_\pi(s) = \nabla \left[\sum_a \pi(a|s) q_\pi(s,a) \right]$$

$$= \sum_a \left[\nabla \pi(a|s) q_\pi(s,a) + \pi(a|s) \nabla q_\pi(s,a) \right].$$

$$\vdots$$

$$\nabla v_\pi(s) = \sum_{x \in S} \sum_{k=0}^{\infty} Pr(s \to x, k, \pi)$$

$$\sum_a \nabla \pi(a|x) q_\pi(x,a)$$

Pr is transitioning probab. from state s to x in k steps under policy π.

$$\nabla J(\theta) = \nabla v_\pi(s_0)$$

$$\propto \sum_s \mu(s) \sum_a \nabla \pi(a|s) q_\pi(s,a)$$