March 23rd (T) YY, TTL.

1. Demo
   - a Photo/Image
   - b Video

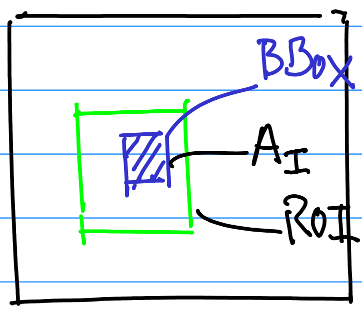
~ rou.py — main Program

~Box.py — Bounding

Testing Program : rectangle_intersection.py

Todo : Send readme.txt,

Threshold T=0.25




Area(Intersection) Blue

$A_I$ v.s. T

$A_I \geq T \to$ Alarm

$A_I < T$ No Action

Define 2 Thresholds
- Upper Threshold $T_u$
- Lower Threshold $T_L$

$A_I$

$T_u \simeq 50\%$

$A_I$ Interested

$T_L \simeq 10\%$

Discarded

$A_I$ is not to be Considered

Discard $A_I$ if it is Bigger than $T_u$

PART II

CTI March 24 (Wed)

DRL-SAC

Soft Actor Critic Algorithm & Apps.

UC Berkeley & Google Pp. 4

Standard RL Objective to maximize Reward

$$\max\left\{\sum_t E_{(s_t,a_t)\sim\rho_\pi}\left[r(\vec{s}_t,\vec{a}_t)\right]\right\} \quad \cdots (1)$$

To Learn Policy

$$\pi(\vec{a}_t|\vec{s}_t) \quad \cdots (2)$$

to maximize Eqn(1).

Augment Eqn(1) with Entropy to maximize entropy at each visited State.

So

$$\alpha \mathcal{H}(\pi(a_t|s_t)) \quad \cdots (3)$$

Weight ("Temperature" parameter determines the relative importance of Entropy term)

Entropy Function

Independent Variable π Policy

Hence Reward in Eqn(1) becomes

$$\max\left\{\sum_t E_{(s_t,a_t)\sim\rho_\pi}\left[r(\vec{s}_t,\vec{a}_t)\right]+\alpha\mathcal{H}\left(\pi(a_t|s_t)\right)\right\}$$

Soft Policy Iteration

Q-value (Reward)

$$Q: S\times A \to \mathbb{R} \quad \ldots (1)$$

Discount factor

$$T^{\pi}Q(\vec{s}_t,\vec{a}_t) \triangleq r(\vec{s}_t,\vec{a}_t)+\gamma E_{\vec{s}_{t+1}\sim p}\left[V(\vec{s}_{t+1})\right] \quad \ldots (1*)$$

Bellman operator for Each Action leads to Reward

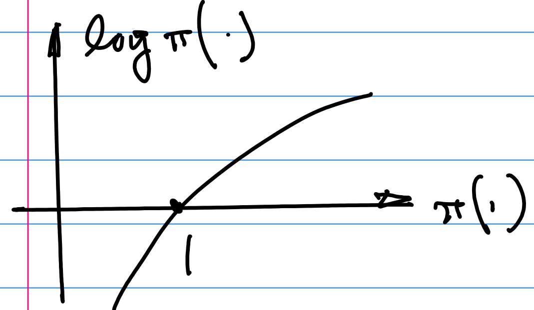
Reward Function

Q-Value function Computed when $\pi(\vec{s}_{t+1}|\vec{a}_{t+1})$ is applied and reward is generated

where

$$V(\vec{s}_{t+1}) = E_{\vec{a}_t\sim\pi}\left[Q(\vec{s}_t,\vec{a}_t)-\alpha\log\pi(\vec{a}_t|\vec{s}_t)\right] \quad \ldots (2)$$

Policy introduced Q-Value (Reward)




Note: Based on (2)

$\alpha\log\pi(\vec{a}_t|\vec{s}_t)$, so $\pi(\vec{a}_t|\vec{s}_t)$ is Reward Based Function

Lemma 1. Bellman Operator T

"0" initial time t=0.

Given initial Condition $Q^0: S\times A\to\mathbb{R}$ with $|A|<\infty$

Define Bellman Operator T as

"All" total Number of finite Actions

$$Q^{k+1}=T^{\pi}Q^{k} \quad \ldots (3)$$

Part II

CTI



Time Index

$$Q^{k+1} = T^{\pi} Q^{k}$$

Q-Value (Reward) Function

Bellman Operator under Policy $\pi$

For $k=0$, we have

$$Q^1 = T^{\pi} Q^0$$

$k=1$, $Q^2 = T^{\pi} Q^1 = T^{\pi}(T^{\pi} Q^0)$

$$= (T^{\pi})^2 Q^0 \quad \ldots (4)$$

...

$k=i$ $\quad Q^{i+1} = (T^{\pi})^{i+1} Q^0 \quad \ldots (5)$

update Q-Value (Reward)

update Policy $\pi$

Note: 1° "Argmin" find minimum from a collection of elements, functions, etc.

Example Argmin $\{1, -1.2, -110, 212\}$ $= -110$.

2° $\pi(\vec{a}_t \mid \vec{s}_t)$ Policy

$\pi(\cdot \mid \vec{s}_t)$ All polices for each/every $a_t$ from $\vec{a}_t$.

$\pi'(\cdot \mid \vec{s}_t)$ one selected from the All polices (all action, e.g. "$\cdot$" for $\vec{a}_t$)

$$\pi_{new} = \underset{\pi' \in \Pi}{\text{argmin}}\, D_{KL}\left(\pi'(\cdot \mid \vec{s}_t) \,\middle\|\, \frac{e^{\frac{1}{\alpha} Q^{\pi_{old}}(\vec{s}_t \mid \cdot)}}{Z^{\pi_{old}}(\vec{s}_t)}\right) \quad \ldots (6)$$

Note: 1° $\pi \in \Pi$

Policy from all collections of the Polices (belongs to) Collection of All Polices

2° $D_{KL}$. KL: Kull Back-Leibler Divergence

3° $\pi'$ One $\pi$ realization from the collection of All Polices, A particular Policy

Generalize Eqn (b), we have

$$\pi_{New} = f\left(\pi_{old} \text{ from one of the all possible Polices} \,\middle|\, \text{Conditions}\right) \quad \dots (6b)$$

$f$ to be minimization problem, so

$$\pi_{New} = \operatorname{argmin} D_{KL}\left(\pi_{old} \text{ from one of the all polices} \,\middle|\, \text{Condition}\right) \quad \dots (6c)$$

Probability or likelihood

Note: Parameterized Policy as Gaussian Distribution

$$\frac{e^{\frac{1}{\alpha} Q^{\pi_{old}}(\vec{s}_t, \cdot)}}{Z^{\pi_{old}}(\vec{s}_t)} \quad \dots (7)$$

Note: In NN Softmax Activation Function we map Neuron Output in $(-\infty, +\infty)$ to Probabilistic distribution $[0, 1]$ e.g.

$$f: z \in (-\infty, +\infty) \rightarrow f(z) \in [0, 1]$$

where

$$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}} \quad \dots (8)$$

Lemma 2.

$$Q^{\pi^*}(\vec{s}_t, \vec{a}_t) \geq Q^{\pi}(\vec{s}_t, \vec{a}_t)$$

where $\pi^*$ is from Eqn (b)

Policy $\pi \in \Pi$, under Eqn (b) converges to $\pi^*$

Note: Eqn (b) needs Better Explanation as why it is formulated as $D_{KL}$ minimization Problem.

Now, introduce parameterized $Q$ function, and Policy $\pi$,

$Q_\theta$, $\pi_\phi$, $\theta$ (Theta), $\phi$ (phi) are

Theta phi

$\phi$: Policy parameters

$$J_Q(\theta) = E_{(\vec{s}_t,\vec{a}_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(\vec{s}_t,\vec{a}_t) - \underbrace{\left(r(\vec{s}_t,\vec{a}_t) + \gamma E_{\vec{s}_{t+1}\sim p}[V_\theta(\vec{s}_{t+1})]\right)}_{\text{Eqn (1*) Bellman Iteration}}\right)^2\right] \quad \ldots (9)$$

Objective Function

$Q(\theta)$ Q-value (Reward) Function with Parameter $\theta$ (Theta)

$$\nabla J_Q(\theta) = E_{(\vec{s}_t,\vec{a}_t)\sim D}\left\{\left[Q_\theta(\vec{s}_t,\vec{a}_t) - \left(r(\vec{s}_t,\vec{a}_t) + \gamma E_{\vec{s}_{t+1}\sim p}[V_\theta(\vec{s}_{t+1})]\right)\right] \cdot \nabla Q_\theta(\vec{s}_t,\vec{a}_t)\right\} \quad \ldots (10)$$

$$J_\pi(\phi) = E_{\vec{s}_t\sim D}\left[E_{\vec{a}\sim\pi_\phi}\left[\alpha\log(\pi_\phi(\vec{a}_t|\vec{s}_t)) - Q_\theta(\vec{s}_t,\vec{a}_t)\right]\right] \quad \ldots (11)$$

$$J_\pi(\phi) = E_{\vec{s}_t\sim D}\left[E_{\vec{a}_t\sim\pi_\phi}[\cdot]\right]$$

For all states    For all actions $\vec{a}_t$

$\pi(\vec{a}_t, \vec{s}_t)$ Eqn (1), PP. 2

Note:

$$\frac{d}{dx}\ln(x) = \frac{1}{x}, \quad \frac{d}{dx}\log_a(x) = \frac{1}{x}\frac{1}{\ln(a)} \quad \ldots (12)$$

$$\vec{a}_t = f_\phi(\epsilon_\pi; \vec{s}_t) \quad \ldots (13)$$

Parameterized Policy? Action

Action Vector multi-dimensional

$$\vec{a}_t = (a_{1t}, a_{2t}, \cdots, a_{Nt})$$

$$J_\pi(\phi) = E_{\vec{s}_t\sim D,\,\epsilon_t\sim N}\left[\alpha\log\pi_\phi(f_\phi(\epsilon_t;\vec{s}_t)|\vec{s}_t) - Q_\theta(\vec{s}_t, f_\phi(\epsilon_t;\vec{s}_t))\right] \quad \ldots (14)$$

Parameter for $\pi$

$\pi_\phi(f_\phi(\epsilon_t|\vec{s}_t)|\vec{s}_t)$ is from

$\pi_\phi(\vec{a}_{\phi t}|\vec{s}_t)$, or is from

$\pi_\phi(\vec{a}_t|\vec{s}_t)$ is from Eqn(2) pp. 2.

$$\hat{\nabla}_\phi J_\pi(\phi) = \nabla_\phi \alpha \log(\pi_\phi(\vec{a}_t|\vec{s}_t)) + (\nabla_{\vec{a}_t} \alpha \log(\pi_\phi(\vec{a}_t|\vec{s}_t)) - \nabla_{\vec{a}_t} Q(\vec{s}_t, \vec{a}_t)) \nabla_\phi f_\phi(\epsilon_t; \vec{s}_t) \quad \ldots (15)$$

Note: Jacobian

Suppose $\vec{f}: \mathbb{R}^n \to \mathbb{R}^m$

$$\vec{x} \in \mathbb{R}^n,\ \vec{f}(\vec{x}) \in \mathbb{R}^m \quad \ldots (16a)$$

Example: $f(x_1, x_2) = f(\vec{x})$, $\vec{x} \in \mathbb{R}^2$

$\vec{x} = (x_1, x_2)$

Then

$$\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x})), \quad \ldots (16b)$$

$$\vec{f}(\vec{x}) \in \mathbb{R}^2. \quad \ldots (16c)$$

Jacobian: Matrix of All its 1st order Partial Derivatives.

$$\vec{J} \triangleq \left[\frac{\partial \vec{f}}{\partial x_1}, \ldots, \frac{\partial \vec{f}}{\partial x_n}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

gradient of $f_1(\vec{x})$ $\quad \ldots (16d)$

$\nabla f_1 =$

$$\nabla f_1(\vec{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \\ \vdots \\ \frac{\partial f_1}{\partial x_n} \end{bmatrix} \quad \ldots (16e)$$

$$\nabla^T f_1(\vec{x}) = \left(\frac{\partial f_1}{\partial x_1} \ \cdots \ \frac{\partial f_1}{\partial x_n}\right) \quad \ldots (16f)$$

Hence, Jacobian

$$\vec{J} = \begin{bmatrix} \nabla^T f_1 \\ \vdots \\ \nabla^T f_m \end{bmatrix} \quad \ldots (16g)$$

Note: $\nabla f_1(\vec{x})$, if $f(\vec{x})$ changed to $f(\vec{x};\phi)$

$\nabla_\phi f_1(\vec{x};\phi)$, so we are taking Partial derivative wrt $\phi$.

$\phi = (\phi_1, \phi_2, \cdots, \phi_n)$, $\vec{\phi}$ is written Simplified as $\phi$.

Automating Entropy Adjustment

$$\max_{\pi_{0:T}} E_{\rho_\pi}\left[\sum_{t=0}^{T} r(\vec{s}_t, \vec{a}_t)\right] \quad \cdots (17)$$

$$\text{s.t. } E_{(\vec{s}_t, \vec{a}_t)\sim\rho_\pi}\left[-\log(\pi_t(\vec{a}_t|\vec{s}_t))\right] \geq \mathcal{H}, \ \forall t \quad \cdots (18)$$

a. $r(\vec{s}_t, \vec{a}_t)$ : reward function

b. $\sum_{t=0}^{T} r(\vec{s}_t, \vec{a}_t)$ : Summation of reward function from $t=0$ to $t=T$;

c. $\max\left[\sum_{t=0}^{T} r(\vec{s}_t, \vec{a}_t)\right]$

maximize the reward function from $t \in [0, T]$ Period.

d. $\max E\left[\sum_{t=0}^{T} r(\vec{s}_t, \vec{a}_t)\right]$

maximize expected reward function from $[0, T]$

Now Add Notation to detail it up

$$\max_{\pi_{0:T}} E_{\rho_\pi}\left[\sum_{t=0}^{T} r(\vec{s}_t, \vec{a}_t)\right]$$

$\pi_{0:T}$ Policy $\pi$ from time 0 to T.

$\rho_\pi$ under Policy, Policy $\pi$

So, we have Eqn (17).

$$-\log(\pi_t(\vec{a}_t|\vec{s}_t)) = \log\frac{1}{\pi_t(\vec{a}_t|\vec{s}_t)}$$

PART II

CTI



## Dynamic Programming, Solving for the policy backward through time

See Ref. pp. 7 "SAC" Paper

Rewrite Objective Function, Eqn (17)

$$\max_{\pi_0}\left\{E\left[r(\vec{s}_0,\vec{a}_0)\right]+\max_{\pi_1}\left\{E[\ldots]+\max_{\pi_T}E\left(r(\vec{s}_t,\vec{a}_t)\right)\right\}\right\}\cdots \quad (18)$$

$$\max_{\pi_0}\left\{E\left[r(\vec{s}_0,\vec{a}_0)\right]+\max_{\pi_1}\left\{E\left[r(\vec{s}_1,\vec{a}_1)\right]+\max_{\pi_2}\left\{E\left[r(\vec{s}_2,\vec{a}_2)\right]\right\}\right\}\right\} \text{ for } t=2$$

$$\max_{\pi_2}\left\{E\left[r(\vec{s}_2,\vec{a}_2)\right]+\max_{\pi_3}\left\{E\left[r(\vec{s}_3,\vec{a}_3)\right]\right\}\right\} \text{ for } t=3$$

$$\max_{\pi_3}\left\{E\left[r(\vec{s}_3,\vec{a}_3)\right)+\max\left\{E\left[r(\vec{s}_4,\vec{a}_4)\right]\right\}\right\} \text{ for } t=4$$

...