ACからDDPG・SACまで

サブタイトルをここに入力

目次

- Actor Critic法
- 確定的方策勾配法
- DDPG法
- Soft Bellman equation
- Soft Actor Critic法

ACTOR CRITIC

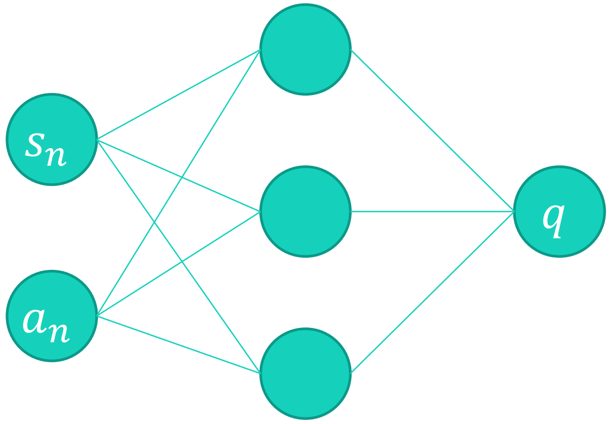
サブタイトルをここに入力

ACTOR CRITIC

- 状態行動価値関数
  - Q Learningでは状態を受け取り、各行動のQ値を返す
    - $f(s_n) = \left(q_{a_1}, q_{a_2}, \dots, q_{a_m}\right)$
    - 行動は離散的である必要あり

	$\boldsymbol{a_1}$	$\boldsymbol{a_2}$	$\boldsymbol{a_3}$
$\boldsymbol{s_1}$	10	20	30
$\boldsymbol{s_2}$	15	10	25
$\boldsymbol{s_3}$	20	15	5

  - Actor Criticでは状態と行動を受け取り、Q値を返す
    - $g(s_n, a_m) = q$
    - この関数gは状態と行動の価値を判定しているためCriticと呼ぶ
    - 連続的な行動も許容



ACTOR CRITC

- 方策関数π
  - Q Learningでは状態に対する各行動のQ値を基にε-greedy法で探索

$$f(s_n) = \left(q_{a_1}, q_{a_2}, \dots, q_{a_m}\right)$$

$$a = \begin{cases} argmax\left(q_{a_1}, q_{a_2}, \dots, q_{a_m}\right)\ for\ 1-\epsilon \\ random(a_1, a_2, \dots, a_m)\ for\ \epsilon \end{cases}$$

  - Actor Criticでは方策関数を直接モデル化
    - この方策関数をactorと呼ぶ
    - 方策関数が確定的な関数の場合、探索のためランダム項を加えることが多い

$$a = \pi(s_n)$$

POINTS

- Actor Critic法は連続的な行動を扱うことができる
- Criticとは状態と行動を受け取り状態行動価値を返す
- Actorとは状態を受け取り行動を返す

DDPG

サブタイトルをここに入力

DDPG

- ここではActor Critic法の実装の一つであるDeep Deterministic Policy Gradient法について紹介する
  - ここでは連続的な行動を仮定したケースについて考える

- DDPG法では状態行動価値関数・方策関数をそれぞれDeep Neural Networkで表現する
  - 状態行動価値関数: $q = Q(s, a)$
  - 方策関数: $a = \pi(s)$

- 探索方法
  - DDPGでは方策関数を決定的な関数として実装している
  - そのため、探索時はランダム性を考慮するためにノイズ項を入れる
    - $a_t^{exploration} = \pi(s_t) + \varepsilon$
  - 探索結果である(s_t, a_t, r_t, s_{t+1})をメモリに保存する点はDQNと同様

DDPG

- 最適化方法
  - Criticの更新はDQNと同様にBellman誤差を減少させるように最適化
    - Bellman誤差: $Q(s_t, a_t) - \left\{r_t + \gamma Q\left(s_{t+1}, a_{t+1}^{exploitation}\right)\right\}$
    - $a_{t+1}^{exploitation}$はノイズ項なしの方策関数から求める $a_{t+1}^{exploition} = \pi(s_{t+1})$
    - 探索と活用で用いる方策が異なることから本手法はoff-policy

  - Actorの更新時はQ値を最大化させるように方策関数を最適化
    - Q値: $Q\left(s_t, \pi(s_t)\right) = Q\left(s_t, a_t^{exploitation}\right)$
    - 実装時はマイナスQ値をlossとして最小化させるように最適化

DDPG

- 実装を見てみる

POINTS

- DDPG法はoff-policyな手法
- CriticとActorをDNNで表現
- 連続的な行動も取り扱い可能

SOFT BELLMAN EQUATION

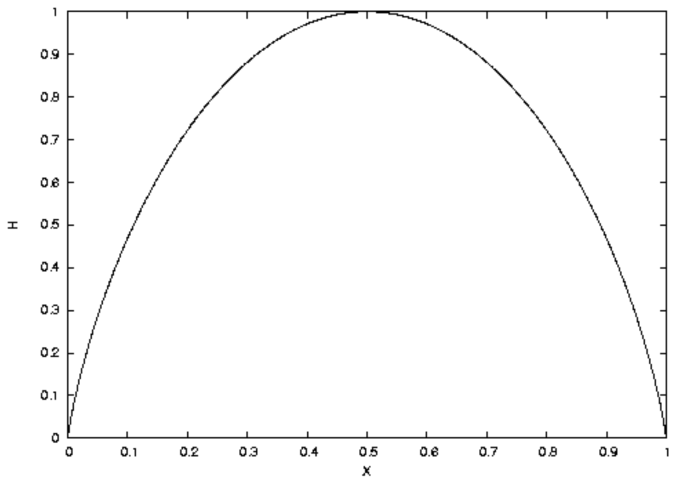
サブタイトルをここに入力

SOFT BELLMAN EQUATION

- エントロピーの定義
  - 本節では確率的方策$\pi(a|s)$について考える。$\pi(a|s)$は状態sでの行動の確率密度関数である
  - このとき方策$\pi(a|s)$に対するエントロピーは次のように定義される
    - 離散エントロピー: $H(\pi|s) = -\sum_a \pi(a|s) \ln \pi(a|s)$
    - 連続エントロピー: $H(\pi|s) = -\int_{-\infty}^{\infty} \pi(a|s) \ln \pi(a|s)\, da = -E[\ln \pi(a|s)]$

- 例1: 右に進む確率をx、左に進む確率を$1-x$とする。このときエントロピーは次の通り
$$H(\pi|s) = -x \ln x - (1-x) \ln(1-x)$$
- 例2: $\pi(a|s)$が平均μ、分散σ^2の正規分布とする。このときエントロピーは次の通り
$$H(\pi|s) = \ln\left(\sigma\sqrt{2\pi e}\right)$$

例1のグラフ



- エントロピーの性質
  - エントロピーは確率分布の平均に依存せず分散にのみ依存
  - 分散が増大するとエントロピーも増大

SOFT BELLMAN EQUATION

- 一般のMDPは割引期待報酬和を最大化する方策を見つけることが目的

$$\max_{\pi} E\left[\sum_{i=0}^{\infty} \gamma^i r(s_{t+i}, a_{t+i}) \,|s_t\right]$$

- Soft MDPでは報酬+エントロピーを最大化する方策を見つける
  - αはエントロピー正則化の定数

$$\max_{\pi} E\left[\sum_{i=0}^{\infty} \gamma^i \left(r(s_{t+i}, a_{t+i}) + \alpha H(\pi|s_{t+i})\right)|s_t\right]$$

- エントロピーは方策のばらつきや不安定性を表しており、不安定な方策であってもより良い報酬を得ることを目的とする
  - →ロバスト性の向上

SOFT BELLMAN EQUATION

- Soft value function

$$
\begin{aligned}
V(s_t) &= E\left[\sum_{i=0}^{\infty} \gamma^i \big(r(s_{t+i}, a_{t+i}) + \alpha H(\pi|s_{t+i})\big) \,|s_t\right] \\
&= E\left[r(s_t, a_t) + \alpha H(\pi|s_t) + \gamma \sum_{i=0}^{\infty} \gamma^i \big(r(s_{t+1+i}, a_{t+1+i}) + \alpha H(\pi|s_{t+1+i})\big) \,|s_t\right] \\
&= E[r(s_t, a_t) + \alpha H(\pi|s_t) + \gamma V(s_{t+1})|s_t]
\end{aligned}
$$

- Soft Q function

$$
\begin{aligned}
Q(s_t, a_t) &= E\left[\sum_{i=0}^{\infty} \gamma^i \big(r(s_{t+i}, a_{t+i}) + \alpha H(\pi|s_{t+i})\big) \,|s_t, a_t\right] \\
&= E[r(s_t, a_t) + \gamma V(s_{t+1})|s_t, a_t] \\
&= E[r(s_t, a_t) + \gamma \{Q(s_{t+1}, a_{t+1}) + \alpha H(\pi|s_{t+1})\}|s_t, a_t]
\end{aligned}
$$

- Q and value function

$$
V(s_t) = E[Q(s_t, a_t) + \alpha H(\pi|s_t)|s_t]
$$

POLICY ITERATION

- 方策関数をsoft q functionのsoftmax関数として定義する

$$\pi(a_t|s_t) = \frac{\exp\left(\frac{1}{\alpha}Q(s_t, a_t)\right)}{\int \exp\left(\frac{1}{\alpha}Q(s_t, a_t)\right)da_t}$$

- 方策関数をvalue functionの式に代入

$$V(s_t) = E\left[Q(s_t, a_t) - \alpha \ln\left(\frac{\exp\left(\frac{1}{\alpha}Q(s_t, a_t)\right)}{\int \exp\left(\frac{1}{\alpha}Q(s_t, a_t)\right)da_t}\right)|s_t\right]$$

$$= E\left[\alpha \ln \int \exp\left(\frac{1}{\alpha}Q(s_t, a_t)\right)da_t\,|s_t\right]$$

- この期待値は行動a_tについて独立なため

$$V(s_t) = \alpha \ln \int \exp\left(\frac{1}{\alpha}Q(s_t, a_t)\right)da_t$$

POLICY ITERATION

- Q値とvalueをそれぞれ次のように更新

$$Q(s_t, a_t) \leftarrow E[r(s_t, a_t) + \gamma V(s_{t+1}) | s_t, a_t]$$

$$V(s_t) \leftarrow \alpha \ln \int \exp\left(\frac{1}{\alpha} Q(s_t, a_t)\right) da_t$$

キャプション

Lorem ipsum dolor sit amet, consectetur adipiscing elit.Etiam eget quam lacus.Vivamus laoreet tempus lacus, Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

REFERENCE

OPEN AI
https://spinningup.openai.com/en/latest/algorithms/sac.html#

SOFT ACTOR-CRITIC: OFF-POLICY MAXIMUM ENTROPY DEEP REINFORCEMENTLEARNING WITH A STOCHASTIC ACTOR
https://arxiv.org/pdf/1801.01290.pdf

LOREM IPSUM DOLOR SIT AMET
Lorem ipsum dolor sit amet consectetuer adipiscing elit, sed diam nonummy nibh euismod

REINFORCEMENT LEARNING WITH DEEP ENERGY-BASED POLICIES
https://arxiv.org/pdf/1702.08165.pdf

MAXIMUM ENTROPY REINFORCEMENT LEARNING (STOCHASTIC CONTROL)
https://www.slideshare.net/DongMinLee32/maximum-entropy-reinforcement-learning-stochastic-control

LOREM IPSUM DOLOR SIT AMET
Lorem ipsum dolor sit amet consectetuer adipiscing elit, sed diam nonummy nibh euismod