

# 講義の概要

- 導入
  - 強化学習の復習
  - 連続値制御と離散値制御
- 1 ステップRLアルゴリズム
  - DDPG, TD3, SAC
- マルチステップRLアルゴリズム
  - マルチステップRLの導入, REINFORCE
  - バイアスとバリエーションのトレードオフ, GAE
  - 方策の単調性能向上, TRPO, PPO
- まとめ & アルゴリズム表
- 補足：
  - 本日紹介する深層強化学習アルゴリズムは「[OpenAI Spinning UP](#)」に実装とともに良くまとまっています
  - 方策勾配の議論は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」が参考になります
  - 参考文献は各ページの下に追記してあります
  - アルゴリズムの実装方法は文献によって違います。本スライドに乗せた実装は一例です。

(h, n) の中身は  
左：方策更新のステップ数  
右：方策評価のステップ数

方策更新の解が**決定的方策**  
(エントロピー正則化なし)

方策更新の解が**確率的方策**  
(エントロピー正則化あり)

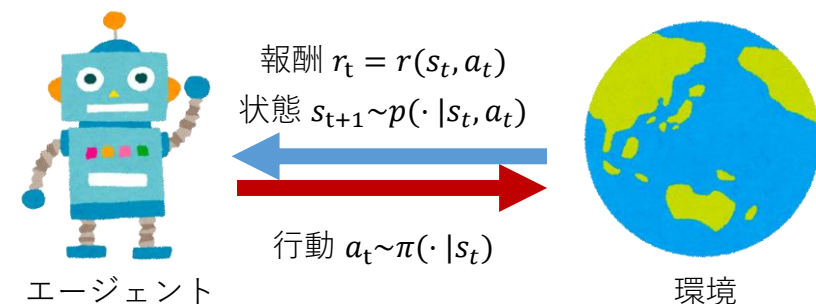
色が濃い部分は  
関数近似&勾配法を使用

		方策更新	方策評価
方策評価を Deep化  方策更新を 連続行動化  工夫3つで 性能向上	Q学習(1, 1)	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
	DQN(1, 1)	$\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\theta-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	DDPG(1, 1)	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\phi-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	TD3(1, 1)	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$
方策評価を Deep化  方策更新を 連続行動化	Soft-Q学習(1, 1)	$\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi \in \Pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') (Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s'))$
	Soft-DQN(1, 1)	$\pi_{\theta}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi \in \Pi} \sum_{a \in A} \pi(s, a) Q_{\theta}(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta-}(a' s') (Q_{\theta-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta-}}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	SAC(1, 1)	$\pi_{\phi} \leftarrow \operatorname{argmin}_{\pi_{\phi}} D_{KL} \left( \pi_{\phi}(\cdot   s) \left\  \frac{\exp(Q_{\theta}(s, a) / \alpha)}{\text{定数}Z} \right\  \right)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_{\phi}(\cdot   s')} [Q_{\theta-}(s', a') - \log \pi_{\phi}(a'   s')] - Q_{\theta}(s, a) \right)^2 \right]$
方策勾配法で 方策更新を置換 & モンテカルロ近似 で方策評価	Q学習(h, n)	$\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1   s_1) (r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi(a_2   s_2) (r(s_2, a_2) + \gamma \dots))$	$Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2   s_2) (r(s_2, a_2) + \gamma \dots)$
	REINFORCE(T, T)	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) Q(s_t, a_t)]$	$Q(s_1, a_1) = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$ 方策評価をモンテカルロ近似
GAEや ベースラインで 性能を向上	Actor-Critic + GAE(T, λ)	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) A(s_t, a_t)]$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$
	TRPO+GAE(T, λ)	$\phi \leftarrow \phi + \alpha \mathbf{H}(\phi)^{-1} \nabla_{\phi'} \mathbf{L}(\phi')$ $\mathbf{L}(\phi') = \sum_{s, a \in S, A} d_h^{\pi_{\phi}}(s) \pi_{\phi'}(a   s) A(s, a)$ $\mathbf{H}(\phi) = \nabla_{\phi'}^2 D_{KL}(\pi_{\phi'} \  \pi_{\phi}) _{\phi'=\phi}$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$

性能の単調向上を  
自然勾配法で実装

# 復習：強化学習の定式化（マルコフ決定過程）

強化学習の目標：マルコフ決定過程（MDP）を解く



MDPの定義

- 行動集合  $A$  : エージェントが選択可能な行動 ( $a \in A$ ) の集合
- 状態集合  $S$  : エージェントは環境の状態 ( $s \in S$ ) に応じて行動を選択する
- 状態遷移確率  $P(\cdot | s_t, a_t): S \rightarrow [0, 1]$  :  $(s_t, a_t)$  から次の状態に遷移する確率
- 報酬関数  $r \in \mathbb{R}^{S \times A}$  : 状態と行動に対する評価 ( $r_t = r(s_t, a_t)$  と略記する)
- 初期状態分布  $p(s_1): S \rightarrow [0, 1]$  : 初期状態  $s_1$  が従う分布
- 割引率  $\gamma \in (0, 1)$  : 目的関数の定義に使うパラメータ

「MDPを解く」とは？ → 何らかの「目的関数を最大化」

MDPの定義は文献によって様々です  
今回はこの定義で説明します

# 復習：強化学習の定式化（収益・目的関数）

「MDPを解く」とは？ → 期待収益を最大化する最適方策  $\pi^*$  の獲得

目的関数関連の表記

- 確率的方策  $\pi(\cdot | s): A \rightarrow [0, 1]$  : エージェントが状態  $s$  で生成する行動が従う分布
- 決定的方策  $\mu(s): S \rightarrow A$  : 特定の行動を返す確率が1の決定的な方策と等価 ( $\pi(a|s) = 1, \pi(a'|s) = 0$ )
- 収益:  $R^\pi = r(s_1, a_1) + \gamma r(s_2, a_2) + \gamma^2 r(s_3, a_3) + \dots \in \mathbb{R}$
- $\pi$  の状態行動価値関数:  $Q^\pi(s, a) = E_\pi[R^\pi | s_1 = s, a_1 = a]$
- $\pi$  の状態価値関数:  $V^\pi(s) = \sum_{a \in A} \pi(s, a) Q^\pi(s, a)$
- 最適方策:  $\pi_* = \operatorname{argmax}_{\pi} \sum_{s \in S} p(s_1) E_\pi[r(s_1, a_1) + \gamma r(s_2, a_2) + \dots]$

このスライドまで：問題設定と目的関数を定義

次：どうやって  $\pi_*$  を学習するか？ → 方策更新と方策評価

目的関数の定義も問題設定によって様々です。  
今回はこの定義で説明します。

# 準備：方策更新と方策評価

## 超重要：方策更新と方策評価

- 方策更新：  $\pi_{k+1}(\cdot|s) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(a|s) Q_k(s, a)$ 
  - モチベーション：  $\pi_k$  よりも良い方策 ( $Q_{\pi_k}(s, a) \leq Q_{\pi_{k+1}}(s, a) \forall (s, a) \in S \times A$ ) が欲しい
  - 実際、  $Q_k = Q_{\pi_k}$  のとき方策更新すると  $Q_{\pi_k} \leq Q_{\pi_{k+1}}$  を満たす
- 方策評価：  $Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \sum_{a' \in A} \pi_{k+1}(a'|s') Q_k(s', a')$ 
  - モチベーション：正しく方策更新するために  $\pi_{k+1}$  の価値  $Q_{\pi_{k+1}}(s, a)$  が欲しい
  - 実際、方策評価を繰り返すと  $Q_k$  は  $k \rightarrow \infty$  で  $Q_{\pi_{k+1}}$  に収束
- なぜ方策更新と方策評価をしたいのか？
  - 更新と評価の繰り返し「 $\dots \rightarrow$  方策更新  $\rightarrow$  方策評価  $\rightarrow$  方策更新  $\rightarrow$  方策評価  $\rightarrow \dots$ 」は  $k \rightarrow \infty$  で  $\pi_k \rightarrow \pi^*$ ,  $Q_k \rightarrow Q^{\pi^*}$  に収束します（証明は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」）

$\pi_{k+1}$  は  $Q_k$  についての貪欲方策：

$$\begin{cases} \pi_{k+1}(a|s) = 1 & \text{if } a = \operatorname{argmax}_{a \in A} Q_k(s, a) \\ \pi_{k+1}(a|s) = 0 & \text{otherwise} \end{cases}$$

今回のスライドでは簡単のために…

- 全状態行動対  $\forall (s, a) \in S \times A$  で更新が行われることにします（探索の話省略）
- RLアルゴリズムとの対応付けのため、価値反復法をQ学習と呼ぶことにします

多くのRLアルゴリズムは方策更新と方策評価を使うとキレイに書けます（Deepでは必要に応じて勾配法で近似）

Q学習（価値反復法）：

- $\pi_{k+1}(\cdot|s) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(a|s) Q_k(s, a)$
- $Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \sum_{a' \in A} \pi_{k+1}(a'|s') Q_k(s', a')$

SARSA：

- $\pi_{k+1} \leftarrow (Q_k \text{ の } \varepsilon\text{-貪欲方策など})$
- $Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) \sum_{a' \in A} \pi_{k+1}(a'|s') Q_k(s', a')$

# 講義の概要

- 導入
  - 強化学習の復習
  - 連続値制御と離散値制御
- 1 ステップRLアルゴリズム
  - DDPG, TD3, SAC
- マルチステップRLアルゴリズム
  - マルチステップRLの導入, REINFORCE
  - バイアスとバリエーションのトレードオフ, GAE
  - 方策の単調性能向上, TRPO, PPO
- まとめ & アルゴリズム表
- 補足：
  - 本日紹介する深層強化学習アルゴリズムは「[OpenAI Spinning UP](#)」に実装とともに良くまとまっています
  - 方策勾配の議論は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」が参考になります
  - 参考文献は各ページの下に追記してあります
  - アルゴリズムの実装方法は文献によって違います。本スライドに乗せた実装は一例です。

# アルゴリズムのまとめ

どのアルゴリズムも「方策更新」と「方策評価」を繰り返しています。

更新と評価のやり方が違うだけです。

	方策更新の解が <b>決定的方策</b> (エントロピー正則化なし)	方策更新の解が <b>確率的方策</b> (エントロピー正則化あり)		色が濃い部分は 関数近似&勾配法を使用

	方策更新	方策評価
方策評価をDeep化	<b>Q学習</b> $\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
方策更新を連続行動化	<b>DQN</b> $\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\theta^-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
工夫3つで性能向上	<b>DDPG</b> $\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\phi^-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	<b>TD3</b> $\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$ <b>二回に一回更新</b>	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i^-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$ <b><math>\mathbf{a}' = \mu_{\phi^-}(s') + \text{ノイズ}</math></b>
方策評価をDeep化	<b>Soft-Q学習</b> $\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') \left( Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s') \right)$
方策更新を連続行動化	<b>Soft-DQN</b> $\pi_{\theta}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_{\theta}(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta^-}(a' s') \left( Q_{\theta^-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta^-}}(s') \right) - Q_{\theta}(s, a) \right)^2 \right]$
	<b>SAC</b> $\pi_{\phi} \leftarrow \operatorname{argmin}_{\pi_{\phi}} D_{KL} \left( \pi_{\phi}(\cdot   s) \left\  \frac{\exp(Q_{\theta}(s, a) / \alpha)}{\text{定数}Z} \right\  \right)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_{\phi}(\cdot   s')} [Q_{\theta^-}(s', a') - \log \pi_{\phi}(a'   s')] - Q_{\theta}(s, a) \right)^2 \right]$

# DDPG : Q学習を連続行動に対応させよう

**Q学習** (方策更新を  $\mu_k: S \rightarrow A$  を使った等価な形で書き換えたもの)

- 方策更新:  $\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$
- 方策評価:  $Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s'|s, a) Q_k(s', \mu_{k+1}(s'))$

**DQN** : Q学習 + **方策評価をNNで近似** (方策更新を  $\mu_\theta: S \rightarrow A$  を使った等価な形で書き換えたもの)

- 方策更新:  $\mu_\theta(s) = \operatorname{argmax}_{a \in A} Q_\theta(s, a)$
- 方策評価:  $Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{(s,a,s',r) \sim \mathcal{D}} \left[ \left( r + \gamma Q_\theta(s', \mu_\theta(s')) - Q_\theta(s, a) \right)^2 \right]$

第一回のDQNと等価です:

$$\left( r + \gamma \max_{a' \in A} Q_\theta(s', a') - Q_\theta(s, a) \right)^2$$

このスライドまで: Q学習とDQNの復習

次: 連続行動の場合に  **$\operatorname{argmax}_{a \in A}$**  はどうする? → 方策更新のNN近似



# DDPG : Deep Deterministic Policy Gradient

- モチベーション :  $\arg \max_a Q_\theta(s, a)$  を別のNN ( $\mu_\phi$ ) で近似しよう  
 $\mu_\phi(s) \approx \operatorname{argmax}_{a \in A} Q_\theta(s, a)$
- やり方 :  $Q_\theta(s, a)$  を最大化する  $a$  を学習すればよい  
→  $Q_\theta(s, \mu_\phi(s))$  を  $\phi$  について勾配上昇法でパラメータ更新しよう

$$\phi \leftarrow \phi + \beta E_{(s, a, s', r) \sim \mathcal{D}} [\nabla_\phi Q_\theta]$$

$$= \phi + \beta E_{(s, a, s', r) \sim \mathcal{D}} \left[ \frac{dQ_\theta}{da}(s, \mu(s)) \frac{d\mu}{d\phi}(s) \right]$$

$$\begin{aligned} \frac{dQ_\theta}{d\phi} &= \frac{dQ_\theta}{da} \frac{da}{d\phi} \\ &= \frac{dQ_\theta}{da}(s, \mu(s)) \frac{d\mu}{d\phi}(s) \end{aligned}$$

**DDPG** : Q学習 + 方策評価をNN ( $Q_\theta$ ) で近似 + **方策更新をNN ( $\mu_\phi$ ) で近似**

- 方策更新 :  $\mu_\phi \leftarrow \operatorname{argmax}_{\mu_\phi} Q_\theta(s, \mu_\phi(s))$
- 方策評価 :  $Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{(s, a, s', r) \sim \mathcal{D}} \left[ \left( r + \gamma Q_\theta(s', \mu_\phi(s')) - Q_\theta(s, a) \right)^2 \right]$

$\mu_\phi$  : アクター  
 $Q_\theta$  : クリティック

# TD3: Twin-Delayed DDPG

モチベーション：  
DDPGの性能を上げたい

やり方：  
工夫を3つ導入

1. Double Q-learningを導入しQ値の過大評価を低減（参考：[Hado, 2010]など）
  - 方策評価： $Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s,a,s',r) \sim \mathcal{D}} \left[ \left( r(s,a) + \gamma \min_{i=1,2} Q_{\theta_i}(s', \mu_{\phi}(s')) - Q_{\theta}(s,a) \right)^2 \right]$
2. 方策の更新頻度を少なくする（2回に1回）
  - アクターの更新が早すぎるとクリティックが追いつけず、収束性が保証できない場合がある（参考：[Kondo, 2003]など）
3. TD学習の $\operatorname{argmax}_a$ にノイズをのせる
  - $\max_a Q(s,a)$  は誤差に弱い（参考：[Bruno, 2015]）。maxを弱めて過適合を防ぐ。

**TD3**：Q学習 + 方策評価をNN ( $Q_{\theta}$ )で近似 + 方策更新をNN ( $\mu_{\phi}$ )で近似 + **工夫3つ**

- 方策更新 (工夫2: 2回に1回)： $\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$     工夫3:  $a' = \operatorname{clip}(\mu_{\phi}(s') + \operatorname{clip}(\varepsilon, -c, c), a_{\text{low}}, a_{\text{high}}), \varepsilon \sim \mathcal{N}(0, \sigma)$
- 方策評価： $Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s,a,s',r) \sim \mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i}(s', \underline{a'}) - Q_{\theta}(s,a) \right)^2 \right]$

工夫1: Double Q-learning

Hado Hasselt, "Double Q-learning," NIPS2010.

Konda, Vijay, and John Tsitsiklis, "On actor-critic algorithms," SIAM2003.

Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist, "Approximate Modified Policy Iteration and its Application to the Game of Tetris," JMLR2015.

Scott Fujimoto, Herke van Hoof, David Meger, "Addressing Function Approximation Error in Actor-Critic Methods," ICML2018.

# ここまでのおさらい



方策更新の解が**決定的方策**  
(エントロピー正則化なし)



色が濃い部分は  
関数近似 & 勾配法を使用

Deep化  
方策更新を  
連続行動化  
工夫3つで  
性能向上

	方策更新	方策評価
Q学習	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
DQN	$\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\theta^-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
DDPG	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\phi^-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
TD3	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i^-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$

このスライドまで：DDPGとTD3を確認

次：TD3は誤差頑健性を工夫3つで向上させたけど、ほかのやり方もある？→エントロピー正則化

# SAC : Soft Actor-Critic の準備

- **モチベーション** : エントロピー正則化RLは誤差に強い (参考 : [Husain 2021] など)
- **やり方** : Soft Q学習 [Haarnoja 2017] を連続値行動に対応させたい

## Soft-Q学習 : Q学習 + **エントロピー正則化**

- 方策更新 :  $\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$
- 方策評価 :  $Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' | s, a) \sum_{a' \in A} \pi_{k+1}(a' | s') (Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s'))$

方策  $\pi$  のエントロピー

$$\mathcal{H}_{\pi}(s) = - \sum_a \pi(s, a) \log \pi(s, a)$$

(次ページで補足)

実はエントロピー正則化された  $\operatorname{argmax}$  の解はsoftmax方策と同じ (参考: [Vieillard 2021] など) :

$$\pi_{k+1}(s, a) = \frac{\exp\left(\frac{1}{\alpha} Q_k(s, a)\right)}{\sum_{a' \in A} \exp\left(\frac{1}{\alpha} Q_k(s, a')\right)}$$

です。行動集合が離散だと簡単に実現可能 (**Soft-DQN**, 省略) ですが、連続だと難しいので…

## **SAC** : softmax方策を連続行動用に修正する

# SAC : Soft Actor-Critic

パラメータ化された  
ガウス分布が良く使われます  
例:  $\mathcal{N}(\mu_\phi, \sigma_\phi)$

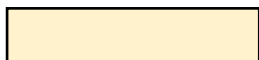
- モチベーション: softmax方策  $\pi(a|s) \propto \exp\left(\frac{1}{\alpha} Q_\theta(s, a)\right)$  を連続分布 ( $\pi_\phi(\cdot|s): A \rightarrow [0, 1]$ ) で近似しよう
- やり方:  $\pi_\phi$  を  $\exp\left(\frac{1}{\alpha} Q_\theta(s, a)\right)$  にKLダイバージェンスの最小化で近づけよう (勾配法で更新)

$$\pi_\phi \leftarrow \operatorname{argmin}_{\pi_\phi} D_{KL} \left( \pi_\phi(\cdot|s) \left\| \frac{\exp(Q_\theta(s, a) / \alpha)}{\text{定数}Z} \right. \right)$$

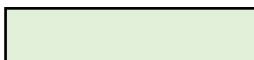
**SAC** : Q学習 + エントロピー正則化 + 方策評価をNN ( $Q_\theta$ ) で近似 + 方策更新をNN ( $\pi_\phi$ ) で近似

- 方策更新:  $\pi_\phi \leftarrow \operatorname{argmin}_{\pi_\phi} D_{KL} \left( \pi_\phi(\cdot|s) \left\| \frac{\exp(Q_\theta(s, a) / \alpha)}{\text{定数}Z} \right. \right)$
- 方策評価:  $Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{(s, a, s', r) \sim \mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_\phi(\cdot|s')} [Q_\theta(s', a') - \log \pi_\phi(a'|s')] - Q_\theta(s, a) \right)^2 \right]$

$\sum_{a \in A} \pi_\phi(a|s') \dots$  のサンプル近似



方策更新の解が**決定的方策**  
(エントロピー正則化なし)



方策更新の解が**確率的方策**  
(エントロピー正則化あり)



色が濃い部分は  
関数近似&勾配法を使用

		方策更新	方策評価
Deep化 方策更新を 連続行動化 工夫3つで 性能向上	Q学習	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
	DQN	$\mu_\theta(s) \leftarrow \operatorname{argmax}_{a \in A} Q_\theta(s, a)$	$Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\theta^-}(s')) - Q_\theta(s, a) \right)^2 \right]$
	DDPG	$\mu_\phi \leftarrow \operatorname{argmax}_{\mu_\phi} Q_\theta(s, \mu_\phi(s))$	$Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\phi^-}(s')) - Q_\theta(s, a) \right)^2 \right]$
	TD3	$\mu_\phi \leftarrow \operatorname{argmax}_{\mu_\phi} Q_\theta(s, \mu_\phi(s))$ <b>二回に一回更新</b>	$Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i^-}(s', \mathbf{a}') - Q_\theta(s, a) \right)^2 \right]$ <b><math>\mathbf{a}' = \mu_{\phi^-}(s') + \text{ノイズ}</math></b>
Deep化 方策更新を 連続行動化	Soft-Q学習	$\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') \left( Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s') \right)$
	Soft-DQN	$\pi_\theta(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_\theta(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta^-}(a' s') \left( Q_{\theta^-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta^-}}(s') \right) - Q_\theta(s, a) \right)^2 \right]$
	SAC	$\pi_\phi \leftarrow \operatorname{argmin}_{\pi_\phi} D_{KL} \left( \pi_\phi(\cdot   s) \left\  \frac{\exp(Q_\theta(s, a) / \alpha)}{\text{定数}Z} \right\  \right)$	$Q_\theta \leftarrow \operatorname{argmin}_{Q_\theta} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_\phi(\cdot   s')} [Q_{\theta^-}(s', a') - \log \pi_\phi(a'   s')] - Q_\theta(s, a) \right)^2 \right]$

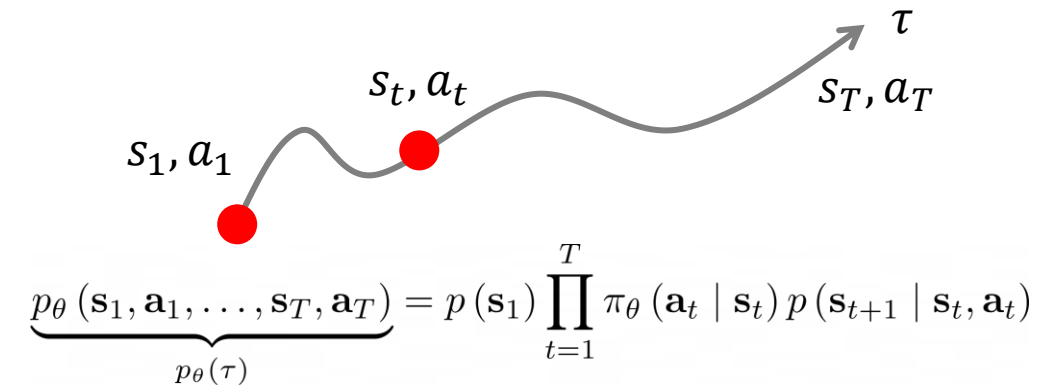
このスライドまで：Q学習ベースの連続値制御RLアルゴリズムを確認

次：方策更新と方策評価を「マルチステップ化」することで、この表をさらに一般化しよう

# 講義の概要

- 導入
  - 強化学習の復習
  - 連続値制御と離散値制御
- 1 ステップRLアルゴリズム
  - DDPG, TD3, SAC
- マルチステップRLアルゴリズム
  - マルチステップRLの導入, REINFORCE
  - バイアスとバリエアンスのトレードオフ, GAE
  - 方策の単調性能向上, TRPO, PPO
- まとめ & アルゴリズム表
- 補足：
  - 本日紹介する深層強化学習アルゴリズムは「[OpenAI Spinning UP](#)」に実装とともに良くまとまっています
  - 方策勾配の議論は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」が参考になります
  - 参考文献は各ページの下に追記してあります
  - アルゴリズムの実装方法は文献によって違います。本スライドに乗せた実装は一例です。

# 準備：軌跡・割引訪問頻度


$$\underbrace{p_{\theta}(s_1, a_1, \dots, s_T, a_T)}_{p_{\theta}(\tau)} = p(s_1) \prod_{t=1}^T \pi_{\theta}(a_t | s_t) p(s_{t+1} | s_t, a_t)$$

## 導出に便利な表記

- ホライゾン  $T$ ：エピソードの長さ。  
割引ありの設定を考えているので  $T = \infty$  で導出をしますが、 $T$  が有限 & 割引なしでも似たような結果が出ます。
- 軌跡  $\tau = \{s_1, a_1, \dots, s_T, a_T\}$ ：  
方策  $\pi$  がたどる状態と行動の軌跡
- 割引訪問頻度：  $d_h^{\pi}(s) = \sum_{t=1}^h \gamma^{t-1} p(s_t = s | \pi)$   
方策  $\pi$  がステップ  $h$  までに状態  $s$  に訪れる割引された総回数の期待値
- 状態価値関数の略記：  $V_{\theta}(s) = \sum_a \pi_{\phi}(s, a) Q_{\theta}(s, a)$   
使っている方策が自明 & 状態行動価値関数が  $Q_{\theta}$  の場合は  $V_{\theta}(s) = \sum_a \pi_{\phi}(s, a) Q_{\theta}(s, a)$  とします



# 復習：Q学習・DQN

## Q学習

- 方策更新： $\pi_{k+1}(s_1, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a_1 \in A} \pi(a_1 | s_1) Q_k(s_1, a_1)$
- 方策評価： $Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2 | s_2) Q_k(s_2, a_2)$

$(s_1, a_1)$ についてだけargmaxで最適化

$(s_1, a_1)$ の報酬の情報だけ使用

$(s_1, a_1)$ についての情報しか使っていないけど...  
もう少し未来の情報を使ってもいいのでは？

このスライドまで： $(s_1, a_1)$ の情報だけ使って更新するアルゴリズム

次： マルチステップの情報 $(s_1, a_1, s_2, a_2, \dots)$ を使うアルゴリズムへ

# 方策更新のマルチステップ化

## 復習：Q学習ベース（DDPGなど）の方策更新

- 方策更新（初期状態分布で書き換えたもの）：

- $\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) Q_k(s_1, a_1)$

$(s_1, a_1)$ についてだけargmaxで最適化

- 最適方策：

- $\pi^* = \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) (r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) (r(s_2, a_2) + \dots))$

$(s_1, a_1, s_2, a_2, \dots)$ の全てをargmaxで最適化

## マルチステップ方策更新 [Efroni 2018]：方策更新をマルチステップにして一般化しよう

1ステップ更新:  $\operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) Q_k(s_1, a_1)$

2ステップ更新:  $\operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) (r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) Q_k(s_2, a_2))$

⋮

hステップ更新:  $\operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) (r(s_1, a_1) + \gamma (\dots + \gamma \sum_{s_h \in S} P(s_h | s_{h-1}, a_{h-1}) \sum_{a_h \in A} \pi(a_h | s_h) Q_k(s_h, a_h)))$

⋮

Tステップ更新:  $\pi^* = \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) (r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) (r(s_2, a_2) + \dots))$

## RL以外のアルゴリズムもマルチステップ貪欲方策で一般化できます

h=1: Q学習, DDPG, SAC, など

h>1: Alpha-Goなど

h=T: モンテカルロ木探索, モデル予測制御, 方策勾配法など

# 方策更新のマルチステップ化と方策勾配法

「方策勾配法」はTステップ（ $\infty$ ステップ）方策更新を勾配法で求めています。導出してみましょう。

**Q学習ベース（DDPGなど）の方策更新**：簡単のために方策評価が完璧に行われた状況を仮定（ $\bar{Q}_{\pi_\phi} = Q_{\pi_\phi}$ を満たす関数）

- 方策更新：  $\pi_\phi \leftarrow \operatorname{argmax}_{\pi_\phi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1)$
- 実装方法：  $J_1(\phi) = \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1)$  として、  
 $\nabla_\phi J_1(\phi) = \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \nabla_\phi \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1)$  について勾配上昇で  $\operatorname{argmax}_{\pi_\phi}$  を計算していた

▼  $\operatorname{argmax}_{\pi_\phi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1)$  を分解してみよう

$\bar{Q}_{\pi_\phi}(s_1, a_1)$  を1ステップ分解

$$\operatorname{argmax}_{\pi_\phi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1)$$

$$\operatorname{argmax}_{\pi_\phi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \left( \underline{r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \bar{Q}_{\pi_\phi}(s_2, a_2)} \right)$$

$$\operatorname{argmax}_{\pi_\phi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \left( r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) (r(s_2, a_2) + \gamma \dots) \right)$$

...

# 方策更新のマルチステップ化と方策勾配法

Q学習ベースの目的関数：  $J_1(\phi) = \sum_{s_1 \in S} p(s_1) \sum_{a \in A} \pi_\phi(a|s_1) Q(s_1, a)$    $\bar{Q}_{\pi_\phi}$  を1ステップ分解

1ステップ分解した目的関数：  $J_2(\phi) = \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \left( r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \bar{Q}_{\pi_\phi}(s_2, a_2) \right)$



$J_2(\phi)$  を最大化してみよう

$\nabla_\phi \pi_\phi(a|s) = \pi_\phi(a|s) \nabla_\phi \log \pi_\phi(a|s)$  と 合成関数の微分 を使って書き換え

$$\begin{aligned} \nabla_\phi J_2(\phi) &= \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \nabla_\phi \log \pi_\phi(a_1|s_1) \left( \underbrace{r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \bar{Q}_{\pi_\phi}(s_2, a_2)}_{= \bar{Q}_{\pi_\phi}(s_1, a_1)} \right) \\ &\quad + \gamma \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \nabla_\phi \log \pi_\phi(a_2|s_2) \bar{Q}_{\pi_\phi}(s_2, a_2) \\ &= \underbrace{\sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \nabla_\phi \log \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1)}_{\text{red underline}} \\ &\quad + \underbrace{\gamma \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \nabla_\phi \log \pi_\phi(a_2|s_2) \bar{Q}_{\pi_\phi}(s_2, a_2)}_{\text{blue underline}} \end{aligned}$$

次ページ：hステップ分解して  $J_h(\phi)$  を考えてみよう

# 方策更新のマルチステップ化と方策勾配法

hステップ分解した目的関数：

$$J_h(\phi) = \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \left( r(s_1, a_1) + \gamma \left( \dots + \gamma \sum_{s_h \in S} P(s_h|s_{h-1}, a_{h-1}) \sum_{a_h \in A} \pi_\phi(a_h|s_{h-1}) \bar{Q}_{\pi_\phi}(s_h, a_h) \right) \right)$$



導出に便利な状態の割引訪問頻度を導入：

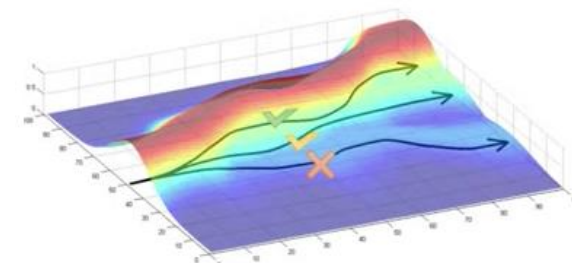
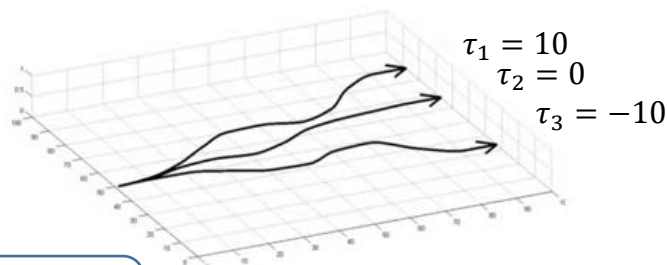
$d_h^\pi(s) = \sum_{t=1}^h \gamma^{t-1} p(s_t = s | \pi)$ ：方策 $\pi$ で状態 $s$ に訪れる割引された総回数の期待値

$$\begin{aligned} \nabla_\phi J_h(\phi) &= \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \nabla_\phi \log \pi_\phi(a_1|s_1) \bar{Q}_{\pi_\phi}(s_1, a_1) \\ &\quad + \gamma \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \nabla_\phi \log \pi_\phi(a_2|s_2) \bar{Q}_{\pi_\phi}(s_2, a_2) \\ &\quad + \gamma^2 \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \sum_{s_2 \in S} P(s_2|s_1, a_1) \sum_{a_2 \in A} \pi_\phi(a_2|s_2) \sum_{s_3 \in S} P(s_3|s_2, a_2) \sum_{a_3 \in A} \pi_\phi(a_3|s_3) \nabla_\phi \log \pi_\phi(a_3|s_3) \bar{Q}_{\pi_\phi}(s_3, a_3) \\ &\quad \vdots \end{aligned}$$

$$= \sum_{s, a \in S, A} d_h^{\pi_\phi}(s) \pi_\phi(a|s) \nabla_\phi \log \pi_\phi(a|s) \bar{Q}_{\pi_\phi}(s, a)$$

$$= E_{\pi_\phi} \left[ \sum_{t=1}^h \gamma^{t-1} \nabla_\phi \log \pi_\phi(a_t|s_t) \bar{Q}_{\pi_\phi}(s_t, a_t) \right]$$

$\infty$ ホライゾンMDPを考えているので割引率が出てきます。  
有限ホライゾンでは $\gamma = 1$ のせいで出てこないことがあります。



# 方策更新のマルチステップ化と方策勾配法

$\nabla_{\phi} J_h(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^h \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) \bar{Q}_{\pi_{\phi}}(s_t, a_t) \right]$  を使って  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_h(\phi)$  がしたい！

## hステップ方策勾配法における方策更新と方策評価

- 方策更新：  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_h(\phi)$ 
  - モチベーション：hステップ貪欲方策を近似したい
  - $\nabla_{\phi} J_h(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^h \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) \bar{Q}_{\pi_{\phi}}(s_t, a_t) \right]$
- 方策評価：  $\bar{Q}_{\pi_{\phi}}(s_t, a_t)$  を何らかの方法で推定
  - モチベーション：方策更新における  $\bar{Q}_{\pi_{\phi}}(s_t, a_t)$  の精度が上がると方策更新の精度も上がる

このスライドまで：方策更新をマルチステップ化して一般化した

次：方策評価をマルチステップ化しよう

# 方策評価のマルチステップ化

## 復習：Q学習ベース（DDPGなど）の方策更新

- 方策評価：

- $Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2 | s_2) Q_k(s_2, a_2)$

報酬は  $r(s_1, a_1)$  だけ使用

- 期待行動価値関数：

- $Q_{\pi_{k+1}}(s_1, a_1) = r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2 | s_2) (r(s_2, a_2) + \dots)$

$(s_1, a_1, s_2, a_2, \dots)$  の全ての報酬を使用

## マルチステップ方策評価 [Kozuno, 2021]：方策評価をマルチステップにして一般化しよう

1ステップ評価：  $Q(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) Q(\mathbf{s}_2, \mathbf{a}_2)$

2ステップ評価：  $Q(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) \left( r(s_2, a_2) + \gamma \sum_{s_3 \in S} P(s_3 | s_2, a_2) \sum_{a_3 \in A} \pi(a_3 | s_3) Q(\mathbf{s}_3, \mathbf{a}_3) \right)$

⋮

nステップ評価：  $Q(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) \left( \dots + \gamma \sum_{s_{n+1} \in S} P(s_{n+1} | s_n, a_n) \sum_{a_{n+1} \in A} \pi(a_{n+1} | s_{n+1}) Q(\mathbf{s}_{n+1}, \mathbf{a}_{n+1}) \right)$

⋮

Tステップ評価：  $Q(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2 | s_1, a_1) \sum_{a_2 \in A} \pi(a_2 | s_2) \left( r(s_2, a_2) + \gamma \sum_{s_3 \in S} P(s_3 | s_2, a_2) \sum_{a_3 \in A} \pi(a_3 | s_3) (\dots) \right)$

## マルチステップ方策評価でいろいろ一般化できます

n=1: Q学習, DDPG, SAC, など

n>1: nステップ収益（Rainbowなどで使用）

n=T: モンテカルロ近似（REINFORCEなどで使用）

# マルチステップ強化学習

## マルチステップ方策更新とマルチステップ方策評価

- **hステップ方策更新**:  $\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1 | s_1) \left( r(s_1, a_1) + \gamma \left( \dots + \gamma \sum_{s_h \in S} P(s_h | s_{h-1}, a_{h-1}) \sum_{a_h \in A} \pi(a_h | s_{h-1}) Q_k(s_h, a_h) \right) \right)$
- **nステップ方策評価**:  $Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \left( \dots + \gamma \sum_{s_{n+1} \in S} P(s_{n+1} | s_n, a_n) \sum_{a_{n+1} \in A} \pi_{k+1}(a_{n+1} | s_{n+1}) Q_k(s_{n+1}, a_{n+1}) \right)$
- **理論保障**:
  - $h \geq 1$  かつ  $n \geq 1$  なら、更新と評価の繰り返し「 $\dots \rightarrow h$ 更新  $\rightarrow n$ 評価  $\rightarrow h$ 更新  $\rightarrow n$ 評価  $\rightarrow \dots$ 」は  $k \rightarrow \infty$  で  $h = 1$  よりも早く  $\pi_k \rightarrow \pi^*$ ,  $Q_k \rightarrow Q^*$  に収束します [Efroni, 2018など]

## 方策更新と方策評価に関数近似を導入

## マルチステップ方策更新とマルチステップ方策評価 + 関数近似

- **hステップ方策更新**:  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_h(\phi)$ 
  - 方策勾配:  $\nabla_{\phi} J_h(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^h \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) \mathbf{Q}(s_t, a_t)]$
- **nステップ方策評価**:  $\mathbf{Q}(s_1, a_1) \leftarrow \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}))_i$
- **Q関数の更新**:  $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} [(Q(s_1, a_2) - Q_{\theta}(s_1, a_1))^2]$

$$\mathbf{V}_{\theta}(s) = \sum_a \pi_{\phi}(s, a) Q_{\theta}(s, a)$$

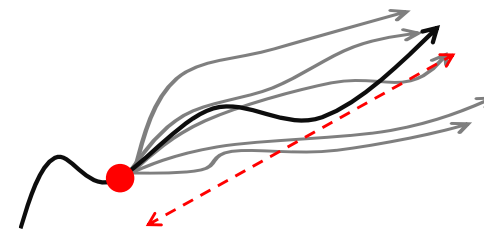
$h = T$  の勾配を求める定理が方策勾配定理と呼ばれるため、  
 $h = T$  のアルゴリズムは「方策勾配法ベースのアルゴリズム」と呼ばれることがあります。



# 方策勾配法ベース ( $h = T, n = T$ ) : REINFORCE

REINFORCE : Tステップ方策勾配 & Tステップ方策評価をモンテカルロ近似

- $h = T$ ステップ方策更新 :  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ 
  - 勾配 :  $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) Q(s_t, a_t)]$
- $n = T$ ステップ方策評価 :  $Q(s_1, a_1) = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$ 
  - N個の軌跡で  $Q \approx Q_{\pi_{\phi}}$  とモンテカルロ近似した形

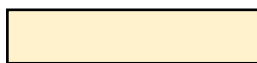


$$Q = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$$

このスライドまで : 「Q学習 ( $h = 1, n = 1$ ) 」 と 「REINFORCE ( $h = T, n = T$ ) 」 の  
極端なアルゴリズムしか見てない !

次 : 中間のアルゴリズム & バイアスとバリエアンスのトレードオフ

(h, n) の中身は  
 左：方策更新のステップ数  
 右：方策評価のステップ数



方策更新の解が**決定的方策**  
 (エントロピー正則化なし)



方策更新の解が**確率的方策**  
 (エントロピー正則化あり)



色が濃い部分は  
関数近似&勾配法を使用

(1, 1)

	方策更新	方策評価
<b>Q学習(1, 1)</b>	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
<b>DQN(1, 1)</b>	$\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\theta^-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
<b>DDPG(1, 1)</b>	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta^-}(s', \mu_{\phi^-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
<b>TD3(1, 1)</b>	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i^-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$
<b>Soft-Q学習(1, 1)</b>	$\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') \left( Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s') \right)$
<b>Soft-DQN(1, 1)</b>	$\pi_{\theta}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_{\theta}(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta^-}(a' s') \left( Q_{\theta^-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta^-}}(s') \right) - Q_{\theta}(s, a) \right)^2 \right]$
<b>SAC(1, 1)</b>	$\pi_{\phi} \leftarrow \operatorname{argmin}_{\pi_{\phi}} D_{KL} \left( \pi_{\phi}(\cdot   s) \left\  \frac{\exp(Q_{\theta}(s, a) / \alpha)}{\text{定数}_Z} \right. \right)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_{\phi}(\cdot   s')} [Q_{\theta^-}(s', a') - \log \pi_{\phi}(a'   s')] - Q_{\theta}(s, a) \right)^2 \right]$
<b>Q学習(h, n)</b>	$\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1   s_1) \left( r(s_1, a_1) + \gamma \left( \dots + \gamma \sum_{s_h \in S} P(s_h   s_{h-1}, a_{h-1}) \sum_{a_h \in A} \pi(a_h   s_{h-1}) Q_k(s_h, a_h) \right) \right)$	$Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \left( \dots + \gamma \sum_{s_{n+1} \in S} P(s_{n+1}   s_n, a_n) \sum_{a_{n+1} \in A} \pi_{k+1}(a_{n+1}   s_{n+1}) Q_k(s_{n+1}, a_{n+1}) \right)$
<b>REINFORCE(T, T)</b>	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) Q(s_t, a_t) \right]$	$Q(s_1, a_1) = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$ <b>方策評価をモンテカルロ近似</b>

方策勾配法で  
 方策更新を置換  
 &  
 モンテカルロ近似  
 で方策評価

# 講義の概要

- 導入
  - 強化学習の復習
  - 連続値制御と離散値制御
- 1 ステップRLアルゴリズム
  - DDPG, TD3, SAC
- マルチステップRLアルゴリズム
  - マルチステップRLの導入, REINFORCE
  - バイアスとバリエーションのトレードオフ, GAE
  - 方策の単調性能向上, TRPO, PPO
- まとめ & アルゴリズム表
- 補足：
  - 本日紹介する深層強化学習アルゴリズムは「[OpenAI Spinning UP](#)」に実装とともに良くまとまっています
  - 方策勾配の議論は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」が参考になります
  - 参考文献は各ページの下に追記してあります
  - アルゴリズムの実装方法は文献によって違います。本スライドに乗せた実装は一例です。

# バイアス・バリエーションのトレードオフ

とりあえず方策評価を例に考えます（方策更新でも似たようなことが言えます）

- （復習）  $n$ ステップ方策評価： $Q(s_1, a_1) \leftarrow \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n V_{\theta}(s_{n+1}, a_{n+1}))_i$ 
  - $Q$ 関数の更新： $Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} [(Q(s_1, a_2) - Q_{\theta}(s_1, a_1))^2]$

バイアス大  
バリエーション小

1ステップ評価： $\frac{1}{N} \sum_{i=1}^N (r_1 + \gamma V_{\theta}(s_2, a_2))_i$

$(s_2, a_2)$ がランダム

2ステップ評価： $\frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 V_{\theta}(s_3, a_3))_i$

$(s_2, a_2, s_3, a_3)$ がランダム

$\vdots$

$n$ ステップ評価： $\frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n V_{\theta}(s_{n+1}, a_{n+1}))_i$

$(s_2, a_2, s_3, a_3, \dots, s_{n+1}, a_{n+1})$ がランダム

$\vdots$

モンテカルロ評価： $\frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$

$(s_2, a_2, s_3, a_3, \dots, s_T, a_T)$ がランダム

バイアス小  
バリエーション大

適当な $n$ を選んでもいいけど…

1~ $T$ ステップ評価の結果全部を混ぜてもよさそう（次ページ： $\lambda$ 方策評価）

# $\lambda$ 方策評価

$\lambda$  方策評価 ( $T = \infty$  のとき)

$n$  が大きいほどバリエーションが大きいので...

大きい  $n$  の重みは小さくして、 $\lambda \in [0, 1]$  の重みですべての評価を混ぜてみよう

$T = \infty$  なら

$1 + \lambda + \lambda^2 + \dots = \frac{1}{1-\lambda}$  なので

正規化する

$(1 - \lambda) \times$

$1 \times$

$\lambda \times$

$\lambda^{n-1} \times$

$\lambda^{T-1} \times$



1ステップ評価:  $r_1 + \gamma V_\theta(s_2, a_2)$

2ステップ評価:  $r_1 + \gamma r_2 + \gamma^2 V_\theta(s_3, a_3)$

$\vdots$

$n$ ステップ評価:  $r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n V_\theta(s_{n+1}, a_{n+1})$

$\vdots$

モンテカルロ評価:  $r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T$

$\lambda = 0 \sim 1$  でバイアスとバリエーションのトレードオフが調整できる

( $\lambda = 0$  で1ステップ評価、 $\lambda = 1$  でモンテカルロ評価)

# 方策勾配のバリエアンスを減らそう：ベースライン

## 方策勾配とλ方策評価

- $h$ ステップ方策更新：  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_h(\phi)$ 
  - 勾配：  $\nabla_{\phi} J_h(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^h \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) \mathbf{Q}(s_t, a_t)]$
- λ方策評価：  $\mathbf{Q}(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} Q_n(s_1, a_1)$ 
  - $n$ ステップ状態価値関数の推定：  $Q_n(s_1, a_1) = (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}))_n$
  - Q関数の更新：  $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} [(Q(s_1, a_1) - Q_{\theta}(s_1, a_1))^2]$



## もっとバリエアンス減らせる？

- 実は  $E_{\pi_{\phi}} [\sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) \mathbf{b}(s_t)] = 0$  なので適当なベースライン関数  $\mathbf{b}: \mathcal{S} \rightarrow \mathcal{R}$  を使って変換可能：
  - $\nabla_{\phi} J_h(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^h \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) (Q(s_t, a_t) - \mathbf{b}(s_t))]$
- 適切な  $\mathbf{b}(s_t)$  を使うと  $\nabla_{\phi} J_T(\phi)$  の推定値のバリエアンスが減らせる
  - 状態価値関数を近似した  $\mathbf{b}(s_t) = V_{\theta}(s_t)$  が良く使われます。
    - $Q_{\pi}(s, a)$  の代わりに **アドバンテージ関数**  $A_{\pi}(s, a) = Q_{\pi}(s, a) - V_{\pi}(s)$  を推定しても方策勾配は等価！
    - アドバンテージ関数を  $\lambda$  ステップ評価しよう！

# GAE: Generalized Advantage Estimation

(復習) :  $\lambda$ 方策評価 :  $Q(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n V_{\theta}(s_{n+1}, a_{n+1}))_n$

## Generalized Advantage Estimation (GAE)

- $Q(s, a)$ の代わりにアドバンテージ関数  $A(s, a) = Q(s, a) - V(s)$  を  $\lambda$  推定しよう !

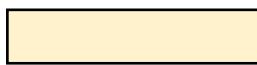
1ステップ評価:  $A_1(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma V_{\theta}(s_2)$

2ステップ評価:  $A_2(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 V_{\theta}(s_3)$

nステップ評価:  $A_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n V_{\theta}(s_{n+1}, a_{n+1})$

- **GAE** :  $A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} A_n(s_1, a_1)$ 
  - Q関数の更新 :  $Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$

(h, n) の中身は  
 左：方策更新のステップ数  
 右：方策評価のステップ数



方策更新の解が**決定的方策**  
 (エントロピー正則化なし)



方策更新の解が**確率的方策**  
 (エントロピー正則化あり)



色が濃い部分は  
関数近似&勾配法を使用

(1, 1)

	方策更新	方策評価
<b>Q学習(1, 1)</b>	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
<b>DQN(1, 1)</b>	$\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\theta-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
<b>DDPG(1, 1)</b>	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\phi-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
<b>TD3(1, 1)</b>	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$
<b>Soft-Q学習(1, 1)</b>	$\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') \left( Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s') \right)$
<b>Soft-DQN(1, 1)</b>	$\pi_{\theta}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_{\theta}(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta-}(a' s') \left( Q_{\theta-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta-}}(s') \right) - Q_{\theta}(s, a) \right)^2 \right]$
<b>SAC(1, 1)</b>	$\pi_{\phi} \leftarrow \operatorname{argmin}_{\pi_{\phi}} D_{KL} \left( \pi_{\phi}(\cdot   s) \left\  \frac{\exp(Q_{\theta}(s, a) / \alpha)}{\text{定数}Z} \right\  \right)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_{\phi}(\cdot   s')} [Q_{\theta-}(s', a') - \log \pi_{\phi}(a'   s')] - Q_{\theta}(s, a) \right)^2 \right]$
<b>Q学習(h, n)</b>	$\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1   s_1) \left( r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi(a_2   s_2) (r(s_2, a_2) + \gamma \dots) \right)$	$Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2   s_2) (r(s_2, a_2) + \gamma \dots)$
<b>REINFORCE(T, T)</b>	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) Q(s_t, a_t) \right]$	$Q(s_1, a_1) = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$ <b>方策評価をモンテカルロ近似</b>
<b>Actor-Critic + GAE(T, λ)</b>	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) A(s_t, a_t) \right]$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$

方策勾配法で  
方策更新を置換  
&  
モンテカルロ近似  
で方策評価

GAEや  
ベースラインで  
性能を向上

今まで  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$  で更新してきたけど... もっと良いの無いの? → TRPO, PPOへ



# 講義の概要

- 導入
  - 強化学習の復習
  - 連続値制御と離散値制御
- 1 ステップRLアルゴリズム
  - DDPG, TD3, SAC
- マルチステップRLアルゴリズム
  - マルチステップRLの導入, REINFORCE
  - バイアスとバリエーションのトレードオフ, GAE
  - 方策の単調性能向上, TRPO, PPO
- まとめ
- 補足：
  - 本日紹介する深層強化学習アルゴリズムは「[OpenAI Spinning UP](#)」に実装とともに良くまとまっています
  - 方策勾配の議論は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」が参考になります
  - 参考文献は各ページの下に追記してあります
  - アルゴリズムの実装方法は文献によって違います。本スライドに乗せた実装は一例です。

# 方策勾配と単調性能向上

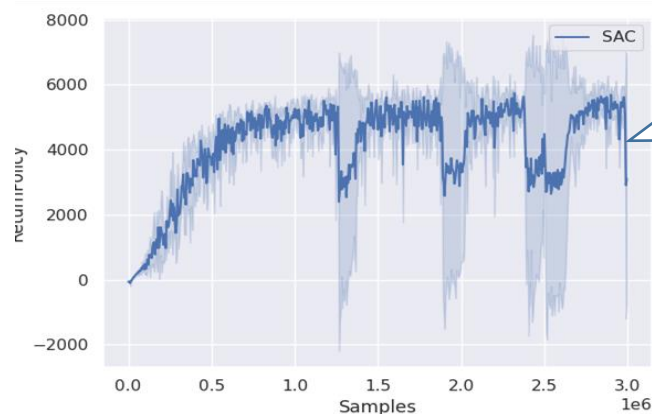
方策勾配による方策の更新をよく見てみよう

- $h$ ステップ方策更新：  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_h(\phi)$ 
  - 勾配：  $\nabla_{\phi} J_h(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^h \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t | s_t) A(s_t, a_t)]$



この勾配をサンプルで近似するわけだけど…

近似や更新の精度が悪いと期待収益( $J_T(\phi)$ )がガタガタすることも：



これは実世界応用では非常に問題！  
例：今まで動いてたロボットが急に動かなくなる

次：性能を単調に向上させたい！→TRPO, PPO

# 方策更新前後の収益の差

方策更新前後の期待収益をよく見てみよう

$$\begin{aligned} J_T(\phi) &= \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi_\phi(a_1|s_1) \left( r(s_1, a_1) + \gamma \left( \dots + \gamma \sum_{s_T \in S} P(s_T|s_{T-1}, a_{T-1}) \sum_{a_T \in A} \pi_\phi(a_T|s_{T-1}) r(s_T, a_T) \right) \right) \\ &= E_{\pi_\phi} [\sum_{t=1}^T \gamma^{t-1} r(s_t, a_t)] = \sum_{s_1 \in S} p(s_1) V_{\pi_\phi}(s_1) \end{aligned}$$

- 何らかの方法で更新された方策を  $\pi_{\phi'}$  とすると、

$$\begin{aligned} J_T(\phi') &= E_{\pi_{\phi'}} [\sum_{t=1}^T \gamma^{t-1} r(s_t, a_t)] \\ &= E_{\pi_{\phi'}} \left[ \sum_{t=1}^T \gamma^{t-1} \left( r(s_t, a_t) + V_{\pi_\phi}(s_t) - V_{\pi_\phi}(s_t) \right) \right] \\ &= E_{\pi_{\phi'}} \left[ \sum_{t=1}^T \gamma^{t-1} \left( r(s_t, a_t) + \gamma V_{\pi_\phi}(s_{t+1}) - V_{\pi_\phi}(s_t) \right) \right] + \sum_{s_1 \in S} p(s_1) V_{\pi_\phi}(s_1) \\ &= E_{\pi_{\phi'}} \left[ \sum_{t=1}^T \gamma^{t-1} A_{\pi_\phi}(s_t) \right] + J_T(\phi) \\ &= \sum_{s,a \in S,A} \mathbf{d}_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_\phi}(s,a) + J_T(\phi) \end{aligned}$$

この値が正なら方策の性能は  
単調に向上するよ！

任意の方策の期待収益の差：  $J_T(\phi') - J_T(\phi) = \sum_{s,a \in S,A} \mathbf{d}_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_\phi}(s,a)$

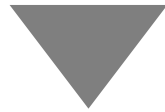
# 方策更新前後の収益の差

この値が正なら方策の性能は  
単調に向上するよ！

任意の方策の期待収益の差： $J_T(\phi') - J_T(\phi) = \sum_{s,a \in S,A} d_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a)$

良く見ると  $\sum_{s,a \in S,A} d_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a)$  は計算できない…

- $d_h^{\pi_{\phi'}}(s)$  は更新後の方策  $\pi_{\phi'}$  が訪れる割引訪問頻度。  
「方策更新の性能向上」を事前に保証したいのに、更新後の方策でデータを集めたら本末転倒！



**頑張って  $d_h^{\pi_{\phi}}(s)$  を使った形に直す**

(証明は[Shulman, 2015]よりもコロンビア大学のIEOR 8100のLecture 7の方が分かりやすいかも)

$$\begin{aligned} J_T(\phi') - J_T(\phi) &= \sum_{s,a \in S,A} d_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a) \\ &\geq \sum_{s,a \in S,A} d_h^{\pi_{\phi}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a) - (\text{定数項}) D_{KL}(\pi_{\phi'} \parallel \pi_{\phi}) \end{aligned}$$

$D_{KL}(\pi_{\phi'} \parallel \pi_{\phi})$  を小さく抑えつつ  $\sum_{s,a \in S,A} d_h^{\pi_{\phi}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a)$  を最大化すれば単調に性能向上しそう！

# 単調な性能向上と自然勾配法

- 今までの方策勾配法：  $\phi \leftarrow \phi + \alpha \nabla_{\phi} J_h(\phi)$

$$\iff \phi \leftarrow \operatorname{argmax}_{\phi'} (\phi' - \phi)^T \nabla_{\phi} J_h(\phi) \text{ s.t. } \|\phi' - \phi\|_2 \leq \varepsilon$$

今まではユークリッドノルムを小さく抑えながら  $J_h(\phi)$  を最大化

- やりたいこと（単調な性能向上）：

- $D_{KL}(\pi_{\phi'} \parallel \pi_{\phi})$  を小さく抑えつつ  $L(\phi') = \sum_{s,a \in S,A} d_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a)$  を最大化

$$\iff \phi \leftarrow \operatorname{argmax}_{\phi'} (\phi' - \phi)^T \nabla_{\phi'} L(\phi') \text{ s.t. } D_{KL}(\pi_{\phi'} \parallel \pi_{\phi}) \leq \varepsilon$$



$(\phi' - \phi)^T \nabla_{\phi'} L(\phi')$  をよく見ると...

$$(\phi' - \phi)^T \nabla_{\phi'} L(\phi') = \left( (\phi' - \phi)^T H(\phi)^{\frac{1}{2}} \right) \left( \nabla_{\phi'} L(\phi')^T H(\phi)^{-\frac{1}{2}} \right)^T \leq \sqrt{\varepsilon} \left\| \nabla_{\phi'} L(\phi')^T H(\phi)^{-\frac{1}{2}} \right\|$$

であり、等号成立は  $\phi' - \phi \propto H(\phi)^{-1} \nabla_{\phi'} L(\phi')$  のとき！

実は  $D_{KL}(\pi_{\phi'} \parallel \pi_{\phi})$  をテイラー展開して二次近似するとフィッシャー情報行列  $H(\phi) = \nabla_{\phi'}^2 D_{KL}(\pi_{\phi'} \parallel \pi_{\phi})|_{\phi'=\phi}$  を使って  $D_{KL}(\pi_{\phi'} \parallel \pi_{\phi}) \approx (\phi' - \phi)^T H(\phi) (\phi' - \phi)$  の形に近似可能

シュワルツの不等式 & KL の制約

## 単調な性能向上のための自然方策勾配

- 方策勾配による更新：  $\phi \leftarrow \phi + \alpha \mathbf{H}(\phi)^{-1} \nabla_{\phi'} \mathbf{L}(\phi')$

自然勾配法と呼ばれます [機械学習のための連続最適化など参照]

- $\mathbf{L}(\phi') = \sum_{s,a \in S,A} d_h^{\pi_{\phi'}}(s) \pi_{\phi'}(a|s) A_{\pi_{\phi}}(s,a)$ ,  $\mathbf{H}(\phi) = \nabla_{\phi'}^2 D_{KL}(\pi_{\phi'} \parallel \pi_{\phi})|_{\phi'=\phi}$

# TRPO: Trust Region Policy Optimization

$$\phi \leftarrow \operatorname{argmax}_{\phi'} L(\phi') \quad \text{---} \quad L(\phi') = E_{\pi_{\phi}} \left[ \sum_{t=1}^h \gamma^{t-1} E_{a_t \sim \pi_{\phi}(\cdot|s_t)} \left[ \frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} A(s_t, a_t) \right] \right]$$

$$\text{s.t. } D_{KL} \left( \pi_{\phi'}(a_t|s_t) \parallel \pi_{\phi}(a_t|s_t) \right) \leq \varepsilon$$

- 制約を破らないように信頼領域法で解く

1. 目的関数・制約条件をTaylor展開し近似

$$\phi' = \operatorname{argmax}_{\phi'} (\phi' - \phi)^T g$$

$$\text{s.t. } \frac{1}{2} (\phi' - \phi)^T H(\phi) (\phi' - \phi) \leq \varepsilon$$

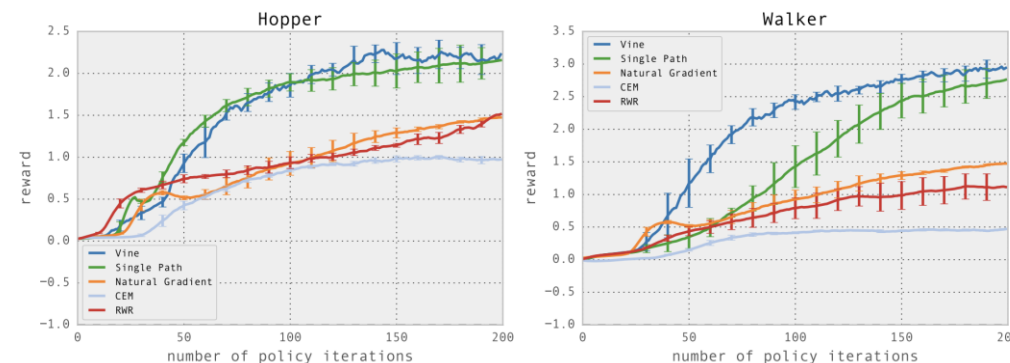
$$\text{where } g = \nabla_{\phi'} L(\phi'), \quad H(\phi) = \nabla_{\phi'}^2 D_{KL} \left( \pi_{\phi'} \parallel \pi_{\phi} \right) \Big|_{\phi'=\phi}$$

2. ラグランジュ未定乗数法で解く

$$\phi' = \phi + \alpha H^{-1} g = \phi + \sqrt{\frac{2\varepsilon}{(H^{-1}g)^T H (H^{-1}g)}} H^{-1} g$$

3. 制約を満たすように線形探索

1. 以下の制約を守っていたら終了
  1.  $\pi_{\phi'}$  がKL制約を破っていないか？
  2.  $L(\phi') - L(\phi) > 0$  ?
2. 満たしていないなら  $\beta \leftarrow \beta/2$



# PPO: Proximal Policy Optimization

$$\phi \leftarrow \operatorname{argmax}_{\phi'} L(\phi') \quad \text{---} \quad L(\phi') = E_{\pi_{\phi}} \left[ \sum_{t=1}^h \gamma^{t-1} E_{a_t \sim \pi_{\phi}(\cdot|s_t)} \left[ \frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} A(s_t, a_t) \right] \right]$$

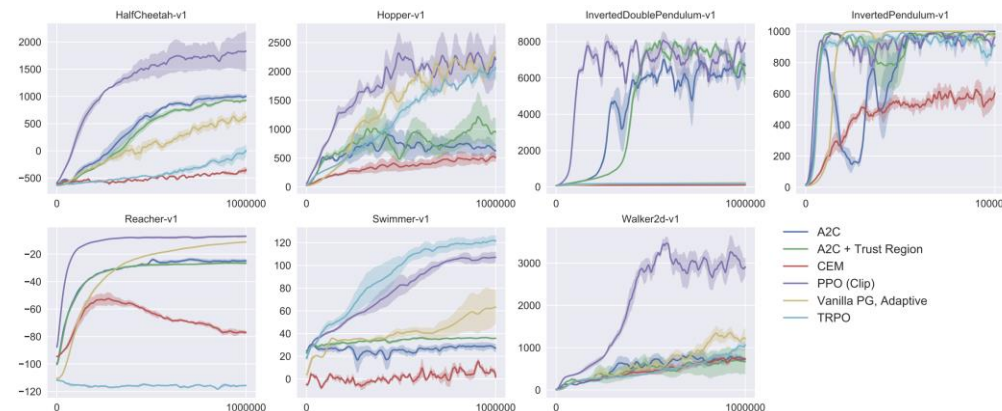
$$\text{s.t. } D_{KL} \left( \pi_{\phi'}(a_t|s_t) \parallel \pi_{\phi}(a_t|s_t) \right) \leq \epsilon$$

- TRPOを実装するのはめんどくさい→簡単な方法で代用しよう (PPO)
- $\frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} A(s_t, a_t)$  は  $A(s_t, a_t) > 0$  なら最適解は  $\frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} \rightarrow \infty$  であり, どんどん方策が離れてしまう。
  - PPO:  $1 - \epsilon < \frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} < 1 + \epsilon$  以外の範囲では  $L(\phi')$  の値が変わらないようにすれば  $\frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} \rightarrow \infty$  が防げそう。

$$L_{PPO}(\phi') = E_{\pi_{\phi}} \left[ \sum_{t=1}^h \gamma^{t-1} E_{a_t \sim \pi_{\phi}(\cdot|s_t)} \left[ \min \left( \frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)} A(s_t, a_t), \operatorname{clip} \left( \frac{\pi_{\phi'}(a_t|s_t)}{\pi_{\phi}(a_t|s_t)}, 1 - \epsilon, 1 + \epsilon \right) A(s_t, a_t) \right) \right] \right]$$

minで下界にする

もとの目的関数



(h, n) の中身は  
左：方策更新のステップ数  
右：方策評価のステップ数

方策更新の解が**決定的方策**  
(エントロピー正則化なし)

方策更新の解が**確率的方策**  
(エントロピー正則化あり)

色が濃い部分は  
関数近似&勾配法を使用

		方策更新	方策評価
方策評価をDeep化	Q学習(1, 1)	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
方策更新を連続行動化	DQN(1, 1)	$\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\theta-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
工夫3つで性能向上	DDPG(1, 1)	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\phi-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	TD3(1, 1)	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$
方策評価をDeep化	Soft-Q学習(1, 1)	$\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') \left( Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s') \right)$
方策更新を連続行動化	Soft-DQN(1, 1)	$\pi_{\theta}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_{\theta}(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta-}(a' s') \left( Q_{\theta-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta-}}(s') \right) - Q_{\theta}(s, a) \right)^2 \right]$
	SAC(1, 1)	$\pi_{\phi} \leftarrow \operatorname{argmin}_{\pi_{\phi}} D_{KL} \left( \pi_{\phi}(\cdot   s) \left\  \frac{\exp(Q_{\theta}(s, a) / \alpha)}{\text{定数}Z} \right\  \right)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_{\phi}(\cdot   s')} [Q_{\theta-}(s', a') - \log \pi_{\phi}(a'   s')] - Q_{\theta}(s, a) \right)^2 \right]$
方策勾配法で方策更新を置換 & モンテカルロ近似で方策評価	Q学習(h, n)	$\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1   s_1) \left( r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi(a_2   s_2) (r(s_2, a_2) + \gamma \dots) \right)$	$Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2   s_2) (r(s_2, a_2) + \gamma \dots)$
	REINFORCE(T, T)	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) Q(s_t, a_t) \right]$	$Q(s_1, a_1) = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$ 方策評価をモンテカルロ近似
GAEやベースラインで性能を向上	Actor-Critic + GAE(T, λ)	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} \left[ \sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) A(s_t, a_t) \right]$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$
性能の単調向上を自然勾配法で実装	TRPO+GAE(T, λ)	$\phi \leftarrow \phi + \alpha \mathbf{H}(\phi)^{-1} \nabla_{\phi'} L(\phi')$ $L(\phi') = \sum_{s, a \in S, A} d_h^{\pi_{\phi}}(s) \pi_{\phi'}(a   s) A(s, a)$ $\mathbf{H}(\phi) = \nabla_{\phi'}^2 D_{KL}(\pi_{\phi'} \  \pi_{\phi}) _{\phi'=\phi}$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$



# 講義の概要

- 導入
  - 強化学習の復習
  - 連続値制御と離散値制御
- 1 ステップRLアルゴリズム
  - DDPG, TD3, SAC
- マルチステップRLアルゴリズム
  - マルチステップRLの導入, REINFORCE
  - バイアスとバリエアンスのトレードオフ, GAE
  - 方策の単調性能向上, TRPO, PPO
- まとめ & アルゴリズム表
- 補足：
  - 本日紹介する深層強化学習アルゴリズムは「[OpenAI Spinning UP](#)」に実装とともに良くまとまっています
  - 方策勾配の議論は「[強化学習（機械学習プロフェッショナルシリーズ）](#)」が参考になります
  - 参考文献は各ページの下に追記してあります
  - アルゴリズムの実装方法は文献によって違います。本スライドに乗せた実装は一例です。

# まとめ

- RLのアルゴリズムの多くは**マルチステップRL**として一般化できます
  - DQN, DDPG, SACは1ステップ方策更新 & 1ステップ方策評価
  - REINFORCEはTステップ方策更新 & Tステップ方策評価
  - GAEは $\lambda$  ステップ方策評価
  - ...
  - その他の中間のアルゴリズムも存在します (Alpha-Goなど)。  
また、まだ未発達のアルゴリズムも存在します ( $\kappa$ ステップ方策更新など)。探してみてください。
- 連続値制御に対応したRLは**方策更新側に関数近似 (アクター)**が入ったアルゴリズムです
- 実は今回のスライドでは、  
1ステップのアルゴリズムは**オフポリシー**  
2ステップ以上のアルゴリズムは**オンポリシー**になっています
  - 2ステップ以上ではアルゴリズムの中に $E_\pi$ が出てくるためです

→次ページ：アルゴリズム表

(h, a) の由来は		方策更新側の関数近似 (アクター)	方策更新側の関数近似 (クリティク)	方策評価側の関数近似 (アクター)	方策評価側の関数近似 (クリティク)
左: 方策更新のステップ数	右: 方策評価のステップ数				
方策更新側	方策評価側				
Q学習 (1, 1)	Q学習 (1, 1)	$Q_{k+1}(s, a) = \arg\max_{a'} Q_k(s, a)$	$Q_{k+1}(s, a) = r(s, a) + \gamma \sum_{s'} P(s' s, a) Q_k(s', a)$		
方策更新側	方策評価側				
DQN (1, 1)	DQN (1, 1)	$Q_k(s) = \arg\max_a Q_k(s, a)$	$Q_k(s) = \arg\max_a E_k \left[ r + \gamma Q_k(s', a) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
DDPG (1, 1)	DDPG (1, 1)	$\mu_k = \arg\max_a \mu_k(s, a)$	$Q_k = \arg\max_a E_k \left[ r + \gamma Q_k(s', a) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
TD3 (1, 1)	TD3 (1, 1)	$\mu_k = \arg\max_a \mu_k(s, a)$	$Q_k = \arg\max_a E_k \left[ r + \gamma \min_{a'} Q_k(s', a) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
SAC (1, 1)	SAC (1, 1)	$\mu_k = \arg\max_a \mu_k(s, a)$	$Q_k = \arg\max_a E_k \left[ r + \gamma \min_{a'} Q_k(s', a) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
Soft-Q学習 (1, 1)	Soft-Q学習 (1, 1)	$Q_{k+1}(s, a) = \arg\max_{a'} \sum_{s'} P(s' s, a) Q_k(s', a) + \alpha V_k(s)$	$Q_{k+1}(s, a) = r(s, a) + \gamma \sum_{s'} P(s' s, a) \sum_{a'} \pi_{k+1}(a' s') \left[ Q_k(s', a') + \alpha V_k(s') \right]$		
方策更新側	方策評価側				
DDPG (1, 1)	DDPG (1, 1)	$\mu_k = \arg\max_a \mu_k(s, a)$	$Q_k = \arg\max_a E_k \left[ r + \gamma \sum_{s'} P(s' s, a) \left( Q_k(s', a) + \alpha V_k(s) \right) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
SAC (1, 1)	SAC (1, 1)	$\mu_k = \arg\max_a \mu_k(s, a)$	$Q_k = \arg\max_a E_k \left[ r + \gamma \sum_{s'} P(s' s, a) \left( Q_k(s', a) + \alpha V_k(s) \right) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
Q学習 (h, a)	Q学習 (h, a)	$Q_{k+1}(s, a) = \arg\max_{a'} \sum_{s'} P(s' s, a) \sum_{a'} \pi_k(a' s') \left[ Q_k(s', a') + \alpha V_k(s) \right]$	$Q_{k+1}(s, a) = r(s, a) + \gamma \sum_{s'} P(s' s, a) \sum_{a'} \pi_k(a' s') \left[ Q_k(s', a') + \alpha V_k(s) \right]$		
方策更新側	方策評価側				
REINFORCE (T, T)	REINFORCE (T, T)	$\phi = \phi + \alpha \nabla_{\theta} J(\phi)$ $\nabla_{\theta} J(\phi) = E_k \left[ \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t s_t) Q(s_t, a_t) \right]$	$Q(s_t, a_t) = \frac{1}{T} \sum_{t=1}^T (r_t + \gamma V_t + \gamma^2 r_{t+1} + \dots + \gamma^{T-t} r_T)$		
方策更新側	方策評価側				
Actor-Critic (T, T)	Actor-Critic (T, T)	$\phi = \phi + \alpha \nabla_{\theta} J(\phi)$ $\nabla_{\theta} J(\phi) = E_k \left[ \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t s_t) A(s_t, a_t) \right]$	$A(s_t, a_t) = (1 - \gamma) \sum_{t=1}^T \gamma^{t-1} A(s_t, a_t)$ $A(s_t, a_t) = -V_k(s_t) + r_t + \gamma V_{k+1}(s_{t+1}) - V_k(s_t)$ $Q_k = \arg\max_a E_k \left[ r + \gamma \sum_{s'} P(s' s, a) \left( Q_k(s', a) + \alpha V_k(s) \right) - Q_k(s, a) \right]$		
方策更新側	方策評価側				
TRPO + GAE (T, T)	TRPO + GAE (T, T)	$\phi = \phi + \alpha \nabla_{\theta} J(\phi)$ $J(\phi) = \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t s_t) A(s_t, a_t)$ $A(s_t, a_t) = E_k \left[ \sum_{t=1}^T \nabla_{\theta} \log \pi_{\theta}(a_t s_t) Q(s_t, a_t) \right]$	$A(s_t, a_t) = (1 - \gamma) \sum_{t=1}^T \gamma^{t-1} A(s_t, a_t)$ $A(s_t, a_t) = -V_k(s_t) + r_t + \gamma V_{k+1}(s_{t+1}) - V_k(s_t)$ $Q_k = \arg\max_a E_k \left[ r + \gamma \sum_{s'} P(s' s, a) \left( Q_k(s', a) + \alpha V_k(s) \right) - Q_k(s, a) \right]$		

(h, n) の中身は  
 左：方策更新のステップ数  
 右：方策評価のステップ数



方策更新の解が**決定的方策**  
 (エントロピー正則化なし)



方策更新の解が**確率的方策**  
 (エントロピー正則化あり)



色が濃い部分は  
 関数近似&勾配法を使用

		方策更新	方策評価
方策評価を Deep化  方策更新を 連続行動化  工夫3つで 性能向上	Q学習(1, 1)	$\mu_{k+1}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_k(s, a)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) Q_k(s', \mu_{k+1}(s'))$
	DQN(1, 1)	$\mu_{\theta}(s) \leftarrow \operatorname{argmax}_{a \in A} Q_{\theta}(s, a)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\theta-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	DDPG(1, 1)	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma Q_{\theta-}(s', \mu_{\phi-}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	TD3(1, 1)	$\mu_{\phi} \leftarrow \operatorname{argmax}_{\mu_{\phi}} Q_{\theta}(s, \mu_{\phi}(s))$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \min_{i=1,2} Q_{\theta_i-}(s', \mathbf{a}') - Q_{\theta}(s, a) \right)^2 \right]$
方策評価を Deep化  方策更新を 連続行動化	Soft-Q学習(1, 1)	$\pi_{k+1}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_k(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{k+1}(s, a) \leftarrow r(s, a) + \gamma \sum_{s' \in S} P(s' s, a) \sum_{a' \in A} \pi_{k+1}(a' s') (Q_k(s', a') + \alpha \mathcal{H}_{\pi_k}(s'))$
	Soft-DQN(1, 1)	$\pi_{\theta}(s, \cdot) \leftarrow \operatorname{argmax}_{\pi} \sum_{a \in A} \pi(s, a) Q_{\theta}(s, a) + \alpha \mathcal{H}_{\pi}(s)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma \sum_{a' \in A} \pi_{\theta-}(a' s') (Q_{\theta-}(s', a') + \alpha \mathcal{H}_{\pi_{\theta-}}(s')) - Q_{\theta}(s, a) \right)^2 \right]$
	SAC(1, 1)	$\pi_{\phi} \leftarrow \operatorname{argmin}_{\pi_{\phi}} D_{KL} \left( \pi_{\phi}(\cdot   s) \left\  \frac{\exp(Q_{\theta}(s, a) / \alpha)}{\text{定数}Z} \right\  \right)$	$Q_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{\mathcal{D}} \left[ \left( r + \gamma E_{a' \sim \pi_{\phi-}(\cdot   s')} [Q_{\theta-}(s', a') - \log \pi_{\phi}(a'   s')] - Q_{\theta}(s, a) \right)^2 \right]$
方策勾配法で 方策更新を置換 & モンテカルロ近似 で方策評価	Q学習(h, n)	$\pi_{k+1} \leftarrow \operatorname{argmax}_{\pi} \sum_{s_1 \in S} p(s_1) \sum_{a_1 \in A} \pi(a_1   s_1) (r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi(a_2   s_2) (r(s_2, a_2) + \gamma \dots))$	$Q_{k+1}(s_1, a_1) \leftarrow r(s_1, a_1) + \gamma \sum_{s_2 \in S} P(s_2   s_1, a_1) \sum_{a_2 \in A} \pi_{k+1}(a_2   s_2) (r(s_2, a_2) + \gamma \dots)$
	REINFORCE(T, T)	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) Q(s_t, a_t)]$	$Q(s_1, a_1) = \frac{1}{N} \sum_{i=1}^N (r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^{T-1} r_T)_i$ 方策評価をモンテカルロ近似
GAEや ベースラインで 性能を向上	Actor-Critic + GAE(T, λ)	$\phi \leftarrow \phi + \alpha \nabla_{\phi} J_T(\phi)$ $\nabla_{\phi} J_T(\phi) = E_{\pi_{\phi}} [\sum_{t=1}^T \gamma^{t-1} \nabla_{\phi} \log \pi_{\phi}(a_t   s_t) A(s_t, a_t)]$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$
	TRPO+GAE(T, λ)	$\phi \leftarrow \phi + \alpha \mathbf{H}(\phi)^{-1} \nabla_{\phi'} \mathbf{L}(\phi')$ $\mathbf{L}(\phi') = \sum_{s, a \in S, A} d_h^{\pi_{\phi}}(s) \pi_{\phi'}(a   s) A(s, a)$ $\mathbf{H}(\phi) = \nabla_{\phi'}^2 D_{KL}(\pi_{\phi'} \  \pi_{\phi}) _{\phi'=\phi}$	$A(s_1, a_1) \leftarrow (1 - \lambda) \sum_{n=1}^T \lambda^{n-1} \mathbf{A}_n(s_1, a_1)$ $\mathbf{A}_n(s_1, a_1) = -V_{\theta}(s_1) + r_1 + \gamma r_2 + \gamma^2 r_3 + \dots + \gamma^n \mathbf{V}_{\theta}(s_{n+1}, a_{n+1})$ $\mathbf{Q}_{\theta} \leftarrow \operatorname{argmin}_{Q_{\theta}} E_{(s_1, a_1) \sim \mathcal{D}} \left[ \left( (A(s_1, a_1) + V_{\theta}(s_1)) - Q_{\theta}(s_1, a_1) \right)^2 \right]$

性能の単調向上を  
自然勾配法で実装