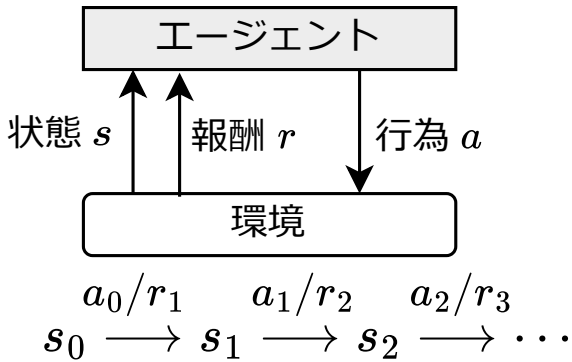
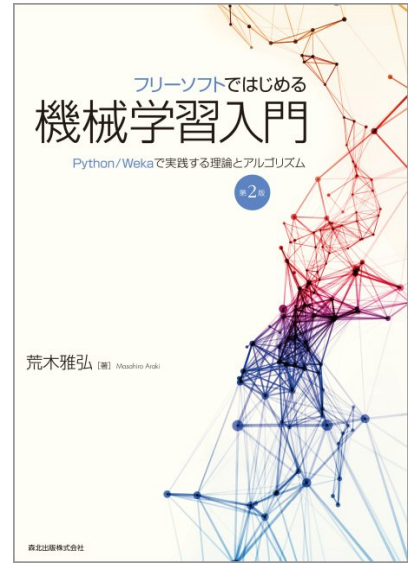
15. 強化学習



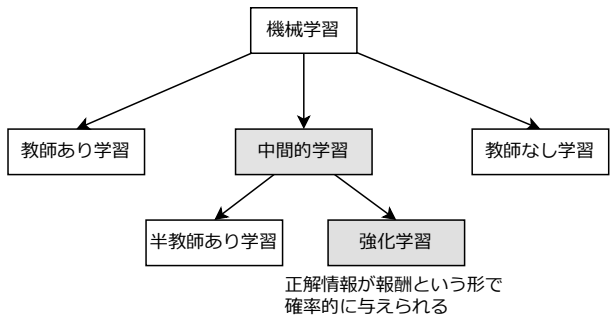
- 15.1 強化学習とは
- 15.2 1状態問題 -K-armed bandit問題-
- 15.3 マルコフ決定過程による定式化
- 15.4 モデルベースの学習
- 15.5 TD学習
- 15.6 部分観測マルコフ決定過程
- 15.7 深層強化学習



- 荒木雅弘:『フリーソフトではじめる機械学習入門(第2版)』(森北出版, 2018年)
- スライドとJupyter notebook
- サポートページ

15.1 強化学習とは (1/2)

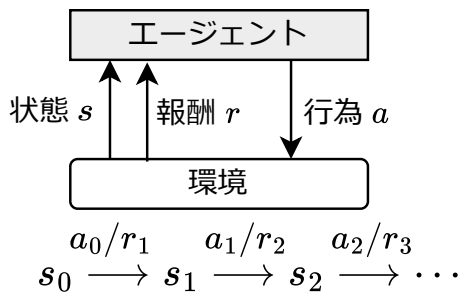
- 強化学習の位置付け：中間的学習
  - 状態変化を伴う環境下で行動するエージェントを想定する
  - 正解（状態に対する正しい行為）は与えられず、時間遅れを伴った報酬（数値）として形を変えて与えられる



- 報酬仮説
  - 学習目標は累積期待報酬最大化で記述できる

15.1 強化学習とは（2/2）

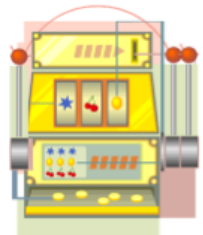
- 強化学習の設定
  - エージェントは環境に対して行為 a_t を行い、環境から行為の結果変化した状態 s_{t+1} と報酬 r_{t+1} を受け取る
  - 時刻 t は離散的に進む
  - エージェントは報酬の期待値が最大となる政策 π（状態から行為への写像）を学習する



- 強化学習の応用例
  - ゲーム、ロボットの制御、対話システム、資源配分 など

15.2 1状態問題 -K-armed bandit問題- (1/3)

- K-armed bandit問題の定義
  - K 本の腕を持つスロットマシンを考える



  - i 番目の腕を引く行為: a_i,　（即時）報酬：$r(a_i)$　$(i = 1, \ldots, K)$
  - 行為の価値: $Q(a_i)$
    - 報酬が確定的な場合
      - すべての a_i を1度ずつ試み、$Q(a_i) = r(a_i)$ が最大となる a_i が最適な行為
    - 報酬が確率的な場合
      - すべての a_i を何度か試み、報酬の平均値 $Q(a_i) = \mathbb{E}(r(a_i))$ が最大となる a_i が最適な行為

15.2 1状態問題 -K-armed bandit問題-（2/3）

- 時刻 t での報酬の平均値 $Q_t(a_i)$ の計算

$$
\begin{aligned}
Q_t(a_i) &= \frac{1}{t}\sum_{j=1}^{t} r_j(a_i) \\
&= \frac{1}{t}\left(r_t(a_i) + \sum_{j=1}^{t-1} r_j(a_i)\right) \\
&= \frac{1}{t}\left(r_t(a_i) + (t-1)Q_{t-1}(a_i)\right) \\
&= Q_{t-1}(a_i) + \frac{1}{t}(r_t(a_i) - Q_{t-1}(a_i))
\end{aligned}
$$

- Q値のインクリメンタルな更新式（更新後の値 ＝ 現在の値 ＋ 学習率 * 誤差）
  - 学習率 η は t の増加に伴って減少させるべきだが、t が大きいときは定数として扱える

$$
Q_{t+1}(a_i) = Q_t(a_i) + \eta(r_{t+1}(a_i) - Q_t(a_i))
$$

15.2 1状態問題 -K-armed bandit問題- (3/3)

- どのように行為 a_i を選ぶか
  - 常に $Q_t(a_i)$ が最大のものを選ぶ方法
    - もっと良い行為があるのに見逃してしまうかもしれない
  - いろいろな a_i を何度も試みる方法
    - 無駄な行為を何度も行ってしまうかもしれない
- ϵ-greedy法
  - 確率 ϵ でランダムに行為を選ぶ
  - 確率 $1-\epsilon$ でその時点においてもっともQ値が高い行為を選ぶ

15.3 マルコフ決定過程による定式化 (1/7)

- K-armed bandit 問題を複数状態の問題に拡張
  - 環境にマルコフ性を仮定
    - 遷移先の状態：直前の状態とそこでの行為のみに依存
    - 報酬：直前の状態と遷移先のみに依存
  - 初期状態から終了状態に至る期間をエピソードとよぶ
    - エピソードの長さが無限の場合もある
  - 目標
    - 1エピソードで得られる報酬の期待値を最大とする政策(状態から行為へのマッピング)の獲得

15.3 マルコフ決定過程による定式化（2/7）

- マルコフ決定過程：状態遷移を伴う問題の定式化
  - 時刻 t における状態: $s_t \in S$
  - 時刻 t における行為: $a_t \in A(s_t)$
  - 報酬 $r_{t+1} \in \mathbb{R}$、確率分布 $p(r_{t+1}|s_t, a_t)$
  - 次状態 $s_{t+1} \in S$、確率分布 $P(s_{t+1}|s_t, a_t)$
  - 価値関数
    - $V^\pi(s_t)$: 状態 s_t から政策 π に従って行動したときに得られる価値
    - $Q(s_t, a_t)$: 状態 s_t における行為 a_t の価値

15.3 マルコフ決定過程による定式化（3/7）

- 問題の具体例：FrozenLake-v1
  - https://gymnasium.farama.org/environments/toy_text/frozen_lake/
  - エージェントは四角に配置されたタイル上で初期状態 S からゴール G を目指して移動する
  - F（Frozen）の状態は歩行可能
    - ただし滑る設定にした場合、意図した方向に移動できないことがある
  - H（Hole）の状態では、穴に落ちてエピソードは終了する
  - 報酬の例
    - G: 1, H: -1

S	F	F	F
F	H	F	H
F	F	F	H
H	F	F	G

15.3 マルコフ決定過程による定式化 (4/7)

- 学習目標
  - 最適政策 π^* の獲得
    - 政策 π:状態から行為へのマッピング
    - 累積報酬の期待値(＝将来の平均)が最大となる政策が最適政策
- 状態価値関数
  - 時刻 t で状態 s_t にいて、その後、政策 π に従って行動したときに得られる累積報酬の期待値
    - γ : 割引率　　$0 \leq \gamma < 1$

$$V^{\pi}(s_t) = \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots) = \mathbb{E}(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i})$$

15.3 マルコフ決定過程による定式化（5/7）

- 1状態先の状態価値関数を用いた定義
  - π^* を $*$ と表記

$$
\begin{aligned}
V^*(s_t) &= \max_{a_t} \mathbb{E}\left(\sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i}\right) \\
&= \max_{a_t} \mathbb{E}\left(r_{t+1} + \gamma \sum_{i=1}^{\infty} \gamma^{i-1} r_{t+i+1}\right) \\
&= \max_{a_t} \mathbb{E}\left(r_{t+1} + \gamma V^*(s_{t+1})\right)
\end{aligned}
$$

15.3 マルコフ決定過程による定式化（6/7）

- 状態遷移確率を明示

$$V^*(s_t) = \max_{a_t}\{\mathbb{E}(r_{t+1}) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)V^*(s_{t+1})\}$$

- VとQとの関係

$$V^*(s_t) = \max_{a_t} Q(s_t, a_t)$$

- Q値による書き換え（ベルマン方程式）

$$Q(s_t, a_t) = \mathbb{E}(r_{t+1}) + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

15.3 マルコフ決定過程による定式化（7/7）

- 強化学習の目標：Q値の推定
  - 環境のモデル（状態遷移確率、報酬の確率分布）が与えられた場合：モデルベースの方法
    - 基本的には動的計画法
  - 環境のモデルが与えられない場合：モデルフリーの方法
    - 得られた報酬に基づき、順次Q値を更新

15.4 モデルベースの学習

- モデルベースのQ値の求め方
  1. 各 s, a に対する Q の推定値 $\hat{Q}(s, a)$ を0に初期化
  2. 現在の状態 s を観測
  3. 以下を繰り返す
     1. 行為 a を選択し実行する
     2. 報酬 r を受け取る
     3. 新しい状態 s' を観測する
     4. $\hat{Q}(s, a)$ を更新する

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a') \quad \gamma : \text{割引率}$$

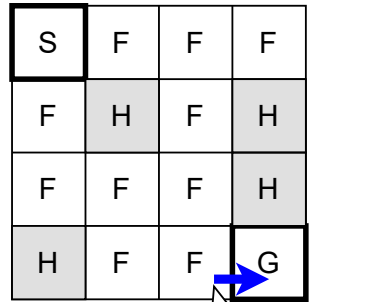
     5. $s \rightarrow s'$

15.5 TD学習 (1/3)

- モデルフリー学習
  - エージェントが探索しながら、得られる報酬に基づいてQ値を更新
- Q値を更新するタイミングに基づく分類
  - エピソードが終了してから更新：モンテカルロ法
  - 一定範囲先の報酬を用いて更新：TD学習
    - TD: Temporal Difference

15.5 TD学習（2/3）

- 報酬と遷移が決定的なTD学習



- 報酬と遷移が決定的な場合のベルマン方程式

$$Q(s_t, a_t) = r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})$$

15.5 TD学習 (3/3)

- 報酬と遷移が確率的なTD学習
  - ベルマン方程式

$$Q(s,a) \leftarrow Q(s,a) + \eta(r + \gamma \max_{a'} Q(s',a') - Q(s,a))$$

- 理論的には、各状態に無限回訪問可能な場合に収束
- 実用的には無限回の訪問は不可能なので、状態推定関数等を用いて、複数の状態を同一とみなす等の工夫が必要

FrozenLake 問題のコーディング（1/4）

- 問題の設定

```
import gymnasium as gym

env = gym.make('FrozenLake-v1', desc=None, map_name="4×4",
               render_mode='ansi', is_slippery=False) #滑らない設定
env.reset()
print(env.render()) #現在の環境を表示
```

```
SFFF
FHFH
FFFH
HFFG
```

FrozenLake 問題のコーディング（2/4）

- TD法によるモデルフリー学習

```
import numpy as np
import matplotlib.pylab as plt

q = np.zeros([N_OBS, N_ACT]) #Q値を0で初期化

#学習時のハイパーパラメータ
EPOCKS = 1000
MAX_ITERATIONS = 100
epsilon = 0.3
gamma = 0.9
eta = 0.9

rewards = np.zeros(EPOCKS) #各エポックでの報酬を記録するarray
```

FrozenLake 問題のコーディング（3/4）

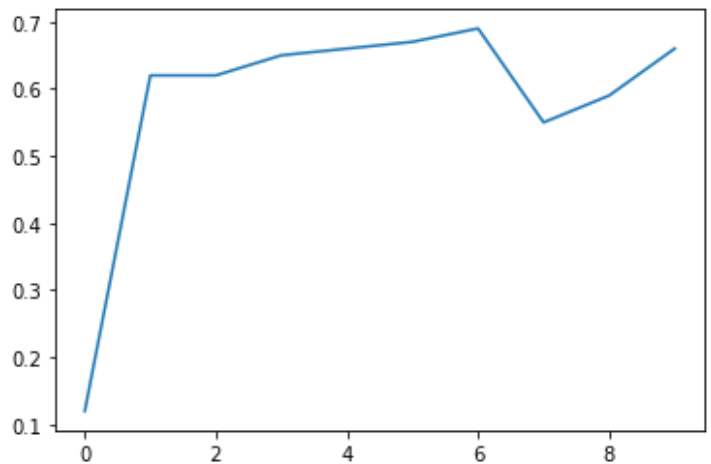
- TD法によるモデルフリー学習

```
for epoch in range(EPOCKS):
    obs = env.reset()[0]
    done = False
    for step in range(MAX_ITERATIONS):
        act = np.argmax(q[obs, :]) #Q値が最大となる行為を求める
        act = np.random.choice(np.where(q[obs, :] == q[obs, act])[0]) #同じ値となるものがあれば、その中からランダムで選択
        if np.random.rand() ⩽ epsilon: #確率epsilonでランダムに行為を選択
            act = env.action_space.sample()
        next_obs, reward, terminated, truncated, info = env.step(act) #行為を実行
        done = terminated or truncated
        if not done: #最終状態（GまたはH）ではないか？
            q[obs, act] += eta * (reward - q[obs, act] + gamma * np.max(q[next_obs, :])) #TD法
        else:
            q[obs, act] += eta * (reward - q[obs, act])
        obs = next_obs
        rewards[epoch] = reward
        if done:
            break
```

FrozenLake 問題のコーディング（4/4）

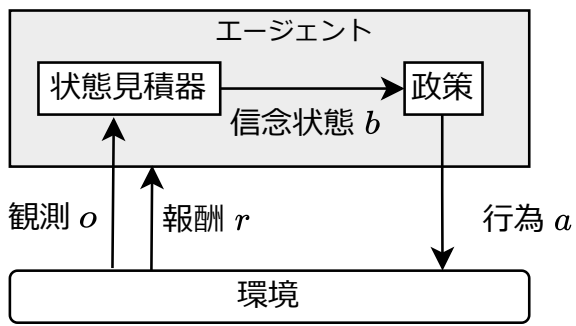
- 100エポック毎の報酬の平均値を求める

```
rates = np.average(rewards.reshape([EPOCKS//100, 100]), axis = 1)
plt.plot(rates)
plt.show()
```



15.6 部分観測マルコフ決定過程

- 部分観測マルコフ決定過程による定式化
  - 状態が確定的には観測できない状況を想定(例:対話システムにおけるユーザの意図)
  - エージェントは、状態の確率分布を信念状態 b_t として持つ
  - 政策に基づいて行為 a_t を行うと、報酬 r_{t+1} と観測 o_{t+1} が確率的に得られる
  - 信念状態 b_t ,行為 a_t ,観測 o_{t+1} から次の信念状態 b_{t+1} を推定する状態見積器 (state estimator) を内部に持つ



15.7 深層強化学習

- 深層学習の強化学習への導入
  - 政策関数による状態価値関数の表現

$$V^{\pi}(s_t) = \sum_{a} \pi(a|s) \times Q(s_t, a_t)$$

  - 価値関数勾配法
    - $Q(s, a)$ の推定にDNNを用いる
    - DNNの学習のための誤差にTD誤差を用いる
  - 方策関数勾配法
    - $\pi(a|s)$ の推定にDNNを用いる
    - V を最大とするようにパラメータを修正する

15.8 まとめ

- 強化学習の問題設定
  - 学習目標は累積期待報酬最大化で記述できるという報酬仮説に基づく
  - 目標は状態から行為を決める関数の獲得だが、正解情報は示されず、遅延した報酬が確率的に得られる
- 最適政策の求め方
  - モデルベース：環境の情報が既知なので線形計画法で解決できる
  - モデルフリー：環境の情報が未知なので繰り返し法を用いる
- 状態数が多い場合など、深層学習との統合が有効