

機械学習（13回目）

創域理工学部 情報計算科学科

桂田 浩一

1

10/19/2023

2

前回の復習

- RNN (Recurrent Neural Network)
 - RNNとは？
 - 従来のRNN
 - LSTM
 - GRU

10/19/2023

3

本日の内容

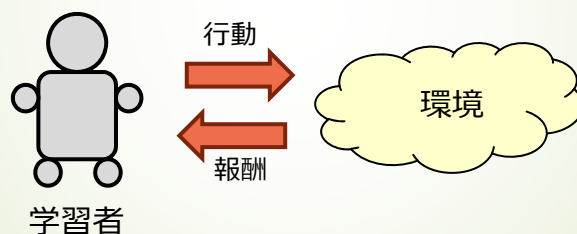
- 強化学習
 - 強化学習とは？
 - Q学習
 - 深層強化学習

10/19/2023

4

強化学習とは？

- 強化学習（Reinforcement Learning）
 - 学習者が環境から得られる報酬に基づき、最適な行動を獲得するような学習を強化学習と呼ぶ。
 - ロボットやエージェントの行動学習等でよく用いられる。

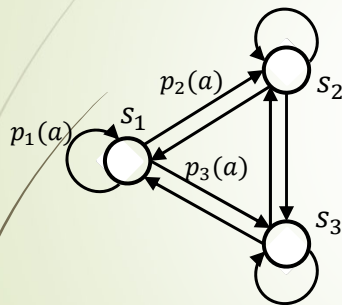


10/19/2023

5


強化学習で前提とする環境・行動

マルコフ決定過程 (Markov Decision Process)



ある状態 s である行動 a をとったとき,
ある確率である状態 s' に遷移する.

$$p(s_t, a_t, s_{t+1}) = p(s_{t+1} = s' | s_t = s, a_t = a)$$

遷移時には報酬 $r(s_t, a_t)$ を受け取る
 (必ずしも正とは限らない.)

10/19/2023

6

状態価値関数と政策, 割引率

政策 (Policy)

- 各状態でどの行動をとるか?
- 状態 s から行動 a への写像
- 通常 π という記号で表される. ($\pi(s) = a$)

状態価値関数

- ある状態 s が政策 π の下でどのくらい価値があるかを表す関数

$$V(s, \pi) = \underbrace{r(s, \pi(s))}_{\text{状態 } s \text{ で行動 } \pi(s) \text{ をとったときの報酬}} + \underbrace{\gamma}_{\text{割引率 } (0 \leq \gamma < 1)} \underbrace{\sum_{s'} p(s, \pi(s), s')}_{\text{状態 } s \text{ で行動 } \pi(s) \text{ をとったとき状態 } s' \text{ になる確率}} \underbrace{V(s', \pi)}_{\text{政策 } \pi \text{ での } s' \text{ の価値}}$$

(一つ先の状態 s' の状態価値は割り引いて考える)

10/19/2023

7

行動価値関数

- 状態 s で行動 a をとった後に, 政策 π に従って行動したときの期待累積報酬
 - $Q_{\pi}(s, a) = r(s, a) + \gamma \sum_{s'} p(s, a, s') V(s', \pi)$

10/19/2023

8

状態価値, 行動価値を最大にする政策

- $V(s, \pi) = r(s, \pi(s)) + \gamma \sum_{s'} p(s, \pi(s), s') V(s', \pi)$
- $Q_{\pi}(s, a) = r(s, a) + \gamma \sum_{s'} p(s, a, s') V(s', \pi)$
- 状態価値を最大にする政策 (最適政策) を π^* とすると,
 - $V(s, \pi^*) = r(s, \pi^*(s)) + \gamma \sum_{s'} p(s, \pi^*(s), s') V(s', \pi^*)$
 $= \max_a \{r(s, a) + \gamma \sum_{s'} p(s, a, s') V(s', \pi^*)\}$
 $= \max_a Q_{\pi^*}(s, a)$
 - $Q_{\pi^*}(s, a) = r(s, a) + \gamma \sum_{s'} p(s, a, s') V(s', \pi^*)$
 $= r(s, a) + \gamma \sum_{s'} p(s, a, s') \max_{a'} Q_{\pi^*}(s', a')$
 $= r(s, a) + \gamma \max_{a'} E[Q_{\pi^*}(s', a')] \quad \leftarrow E[x] : x \text{ の期待値}$



状態 s で行動 a をとったときの報酬
 + 遷移先の s' で最適行動を行った時の期待報酬

10/19/2023

9

Q学習 (Watkins'89)

■ π^* を求めるために、各状態・行動のQ値を更新していく方法

1. 各状態 s , 行動 a について $Q(s, a)$ を初期化, $t = 0$ とする.
2. Q値が収束していれば終了. そうでなければ 3. へ.
3. 現在の状態 s_t で行動 a_t を選ぶ. その結果状態が s_{t+1} になり, 報酬 $r(s_t, a_t)$ を受け取ったとする.
4. $Q(s_t, a_t) \leftarrow (1 - \alpha_t)Q(s_t, a_t) + \alpha_t(r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}))$
5. 2. へ戻る.

期待値の代わりに現在のQ値を使う

以上のようにQ値を更新すると, $\sum_{t=0}^{\infty} \alpha_t = \infty$ かつ $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$ の条件下でQ値が $Q_{\pi^*}(s, a)$ に収束する.

10/19/2023

10

学習時 (ステップ3.) の行動の選び方

■ ランダム法

- ランダムに行動を選ぶ
- 収束は早いが学習途中の報酬は少ない可能性がある

■ グリーディ法

- 常にQ値が最大の行動を選ぶ
- 最適の行動を獲得できない可能性がある

■ ϵ -グリーディ法

- ϵ の確率でランダム法を, $(1 - \epsilon)$ の確率でグリーディ法を行う
- 学習中の報酬もそこそこ得つつ, 最適行動の獲得も保証される

10/19/2023

11

深層強化学習：Deep Q Network

■ Q学習の最適政策を深層学習で求める方法

■ Q学習のステップ4.

$$\begin{aligned}
 Q(s_t, a_t) &\leftarrow \\
 &(1 - \alpha_t)Q(s_t, a_t) + \alpha_t(r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1})) \\
 &= Q(s_t, a_t) + \alpha_t \underbrace{(r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))}_{\text{TD(Temporal Difference)誤差}}
 \end{aligned}$$

■ 誤差の最小化にニューラルネットを用いる

$$\frac{1}{2} (r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))^2$$

を誤差関数とするニューラルネットを作り、誤差を最小化する

10/19/2023

12


APV-MCTS (非同期政策・価値モンテカルロ木探索)

■ α-碁で使われた探索法

■ 深層強化学習 + モンテカルロ法 + セルフプレイ

- モンテカルロ法： ミニマックス法のような手法でなく、確率的に手を選択する。
- セルフプレイ： 自分同士で対戦して強化する
- 政策ネットワーク, 価値ネットワーク： CNN+ReLU (13~15層)

10/19/2023



出題予定の演習課題

- Q学習関係