強化学習

8月13日(土) WSL勉強会

宗政一舟

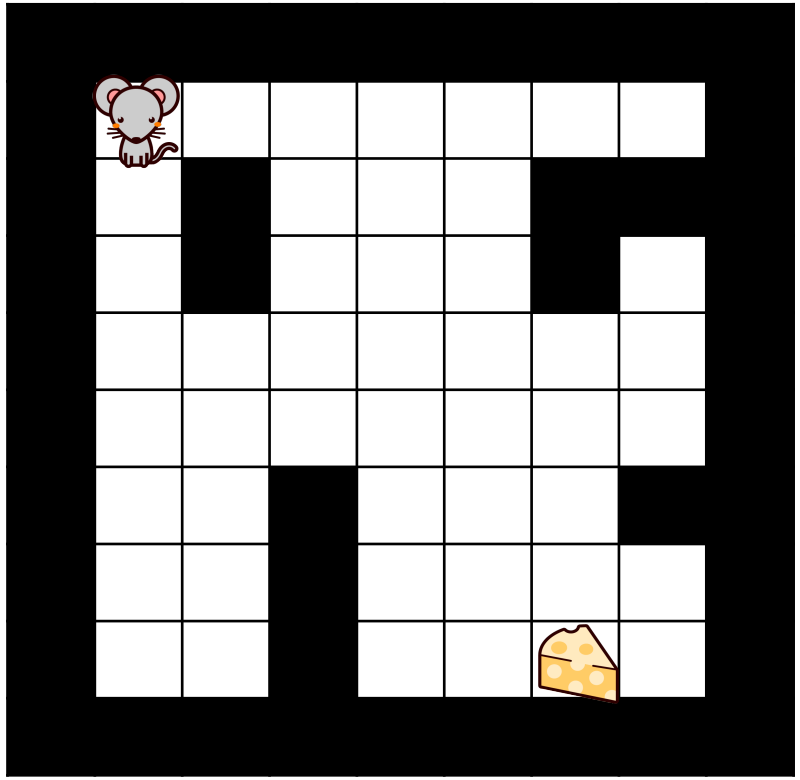
もくじ

1. 今回考える問題
2. 強化学習とは
3. 用語の確認
4. 最適行動価値関数
5. 基本的な解法
6. 実際に解いてみよう

今回考える問題

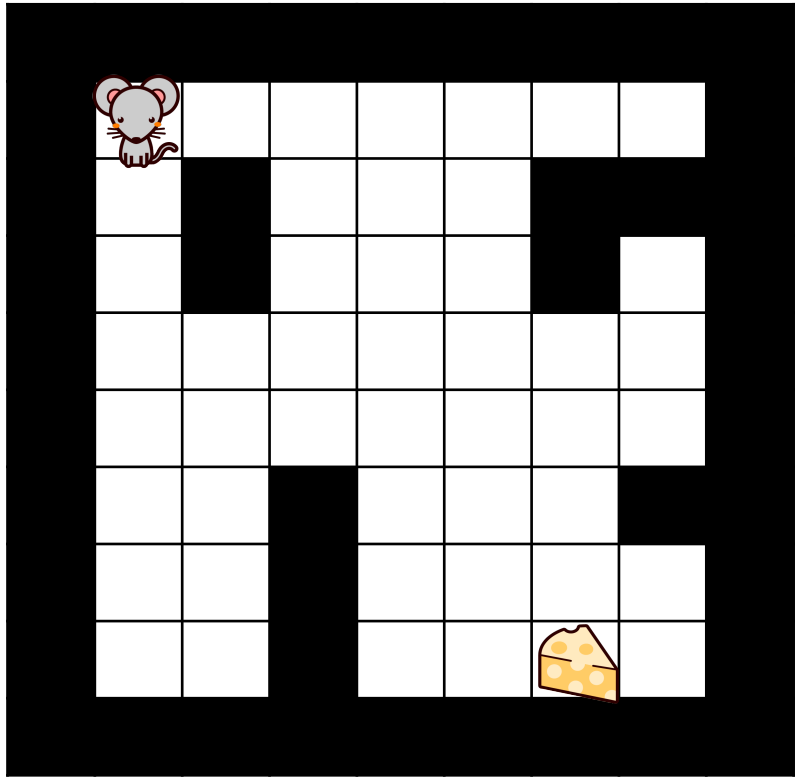
今回考える問題

ネズミがチーズまでたどり着く道のりを学習せよ



今回考える問題

強化学習を用いることにする



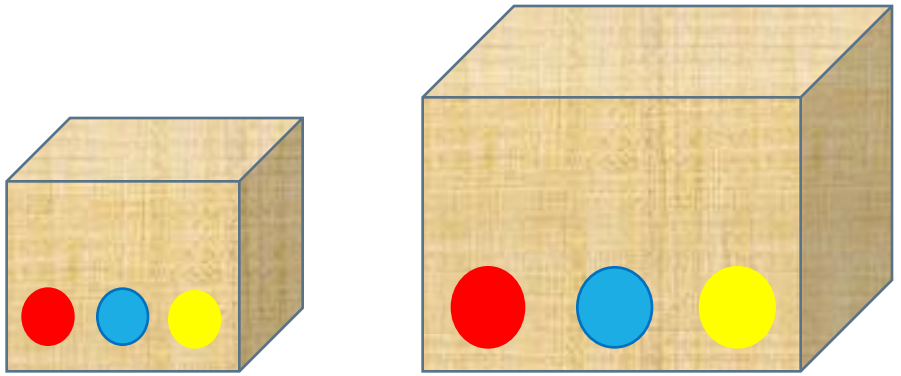
2.強化学習とは

強化学習とは

サルが檻の中に一匹います

サルの前に大きな箱と小さな箱があります

それらの箱には「赤、青、黄」のボタンが付いています

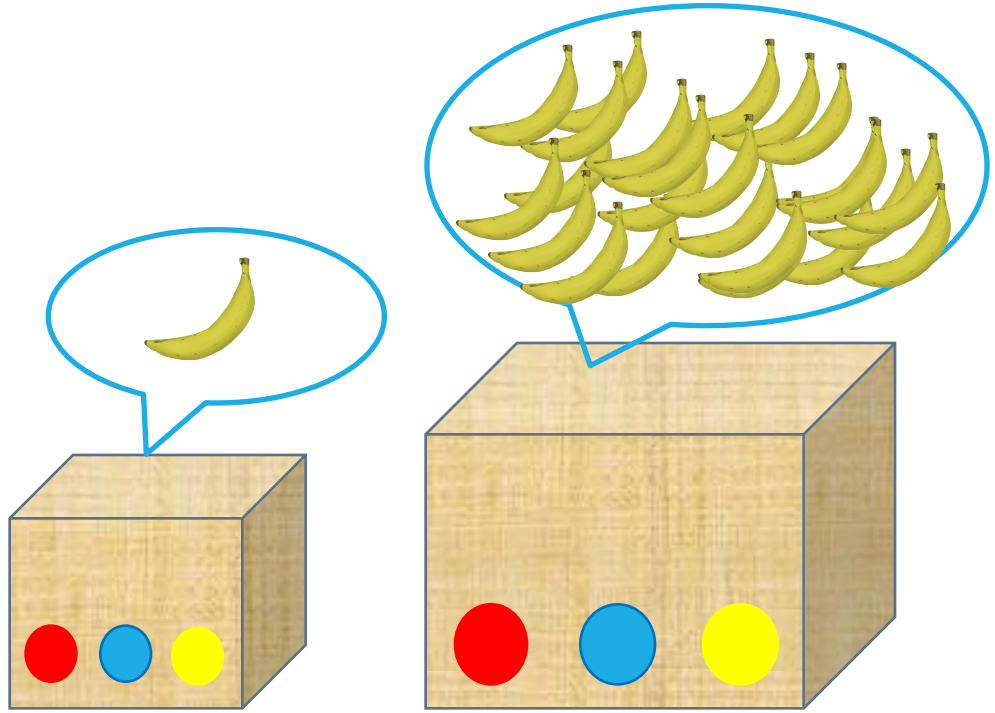


強化学習とは

ちなみに

「小さい箱」にはバナナが1本

「大きい箱」にはバナナがたくさん

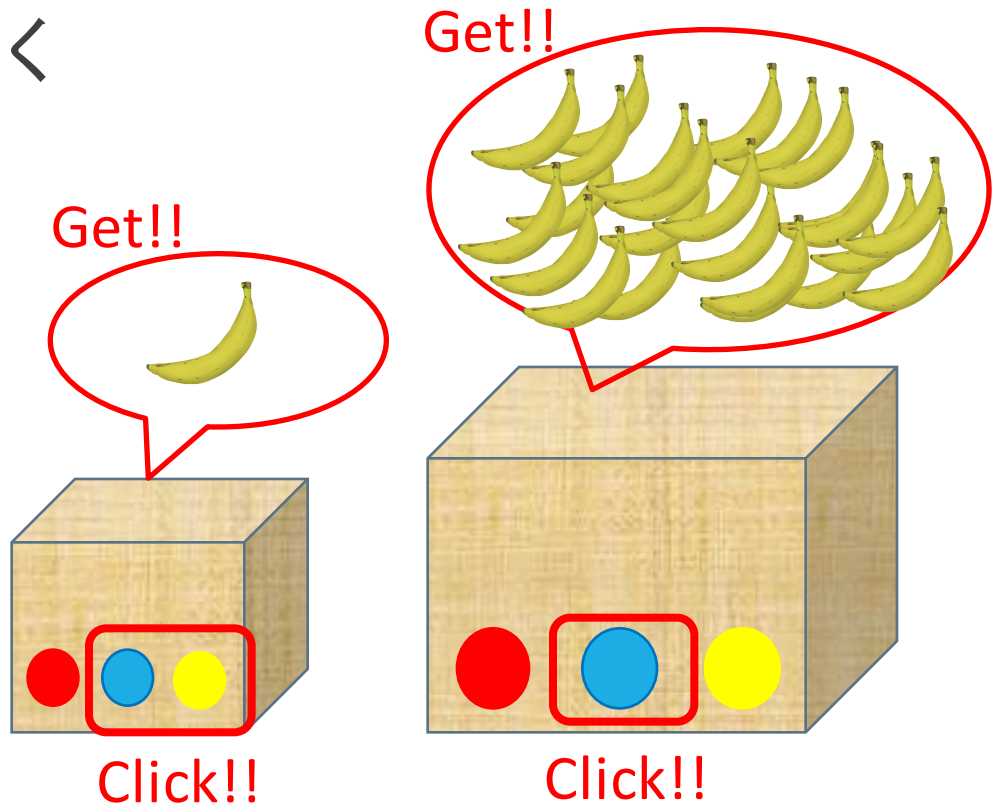


強化学習とは

それぞれの箱は

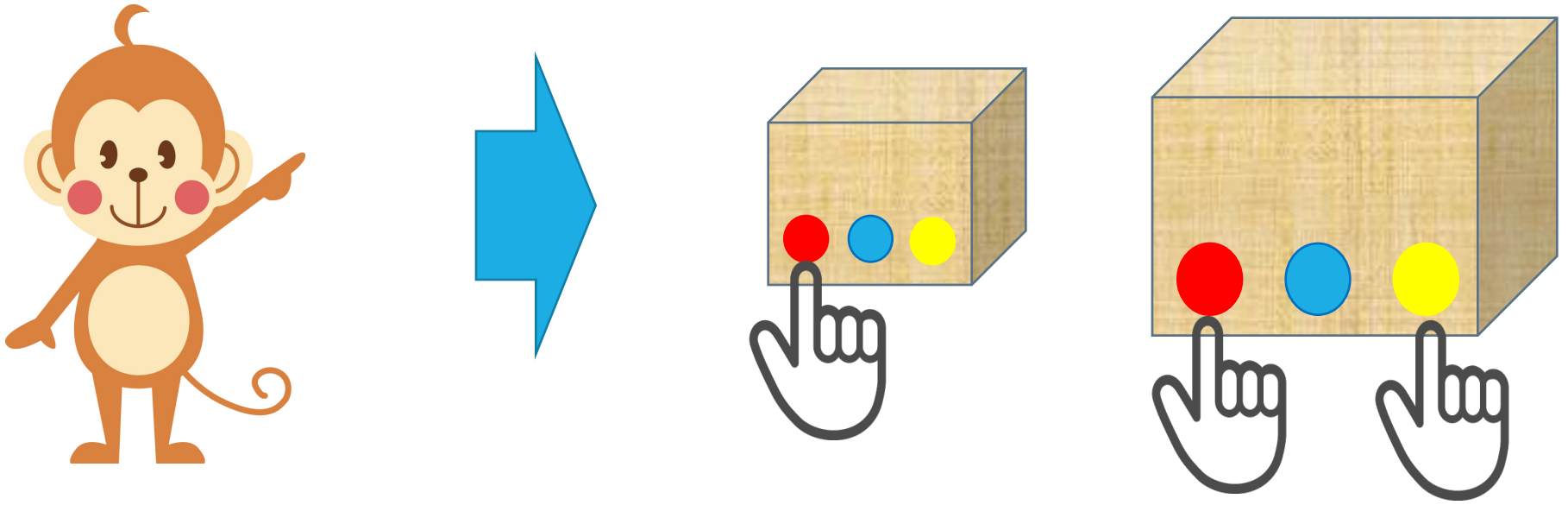
「小さい箱」: 青or黄のボタンを押すと開く

「大きい箱」: 青のボタンを押すと開く



強化学習とは

ただし、それ以外を押してしまうと、、、



強化学習とは

みなさん、ご存知「電撃」をくらいます

強化学習とは

それでもサルは

あきらめずに「大・小」の箱→「赤・黄・青」のボタンを選び続けます

①大きな箱→黄色のボタン→電撃

②小さい箱→青色のボタン→バナナ1本

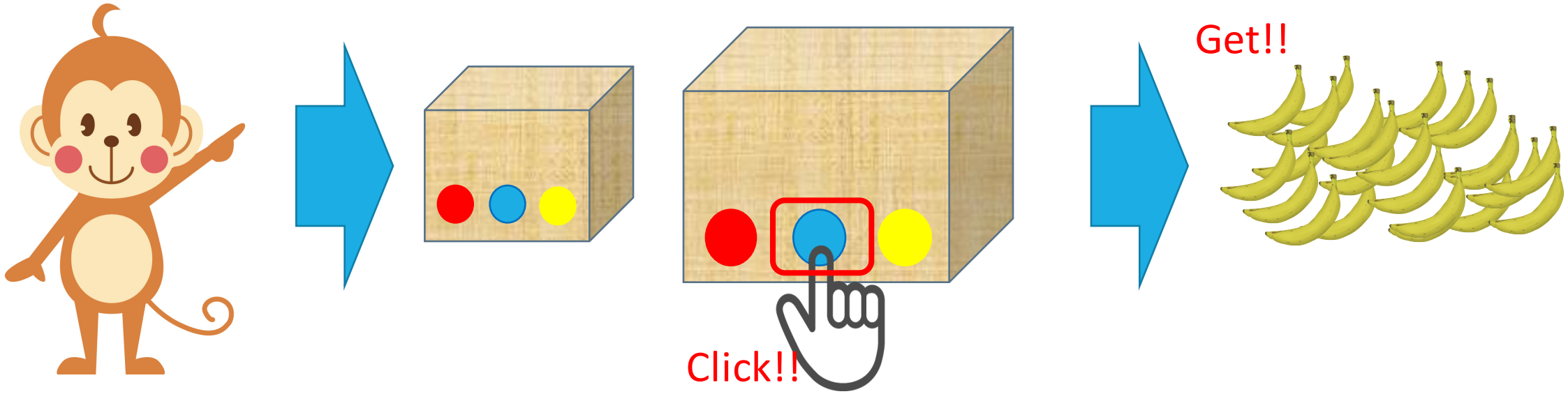
③大きな箱→青色のボタン→大量のバナナ

といった具合で行動を繰り返します

強化学習とは

やがてサルは

大きな箱を選び、青色のボタンを押すことで大量のバナナを得られることを学びます



強化学習とは

- ココでのポイント
  1. 「報酬や罰」は与えられるが正解は教えられない
  2. 「報酬や罰」と「行動」を結びつけて「行動の価値」を評価
  3. 「より報酬が得られそうな行動」を繰り返す

強化学習の基本的な考え方

3. 用語の確認

用語の確認

- 「エージェント」
  - 学習したり行動をしたりする主体のこと
- 「環境」
  - エージェントが行動した結果、状態を変えたり、報酬/罰を与えるもの

- 先ほどの例だと、、
  - 「エージェント」:サル　「環境」:檻の中

用語の確認 (「環境」側)

- 環境はいくつかの「状態」をもつ
  - 例) 檻の中の「状態」をSとする

$S = \{$"初期状態","小さな箱が選ばれている",
"大きな箱が選ばれている","小さな箱が開けられる",
"大きな箱が開けられる","箱を開けるのに失敗"$\}$

- 環境の「状態」はエージェントが「行動」を行うと変化する

時間のステップ$t = 0,1,2 \ldots$
$s_0 =$ "初期状態"
$s_1 =$ "小さな箱が選ばれている" or $s_1 =$ "大きな箱が選ばれている"

用語の確認 (「エージェント」側)

- 環境の「状態」がs_tであるときエージェントが選択することができる「行動」は限られる
- 「状態」がs_tであるとき、エージェントが選択できる行動集合を$\mathrm{A}(s_t)$と書く
  - 例)サルが行うことができる「行動」

$$A(\text{"初期状態"}) = \{\text{"小さな箱を選ぶ"}, \text{"大きな箱を選ぶ"}\}$$
$$A(\text{"小さな箱が選ばれている"}) = \{\text{"赤色"}, \text{"黄色"}, \text{"青色"}\}$$

用語の確認

- エージェントが「状態」s_tで行った「行動」を$a_t(\in A(s_t))$とすると
- 次の「状態」であるs_{t+1}が決まる
- そして「報酬」$r_{t+1}(\in R)$を受け取ることになる
  - 例) 先ほどの檻の中のサル

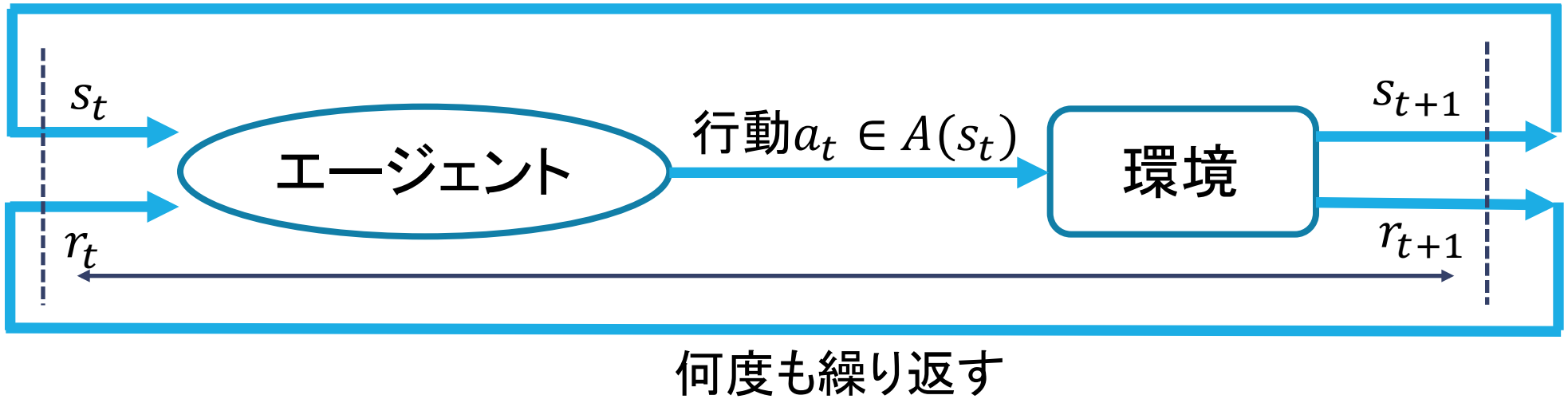
$$S_0 = \text{"初期状態"}$$

$$S_1 = \begin{cases} \text{小箱が選ばれている}(a_0\text{: 小箱を選ぶ}) \\ \text{大箱が選ばれている}(a_0\text{: 大箱を選ぶ}) \end{cases}$$

$$r_1 = 0$$

用語の確認

- 「状態」s_tで「行動」a_tを選択した時の次の「状態」s_{t+1}と「報酬」r_{t+1}は環境によって決定される
- エージェントが学習していくことで変わるもの
  - エージェントの「行動」の選び方→「方策」πとする



用語の確認

- 「状態」s_tの時エージェントが「行動」aを選ぶ確率(方策)を$\pi_t(s,a)$とする
- 強化学習はそれぞれの状態における「方策」$\pi_t(s,a)$を改善していくことを目的としている
- エージェントが「方策」を改善していく方法
  - 行動の価値を表現する行動価値関数を持っていて、最適な価値関数(最適行動価値関数)を求めることで「方策」を改善する

4. 最適行動価値関数

最適行動価値関数

- ある「状態」sにおいて最適な「方策」π_s^*を求める

$$\pi_s^* = argmax_{a \in A(s)} \underline{Q^*(s,a)} \cdots (1)$$

最適行動価値関数

- 最適行動価値関数とは
  - 状態sにおいて行動aを取ることにより、最終的にどれほどの報酬に繋がるのかを考える期待値
- それぞれの状態には行動の数だけ最適価値関数が存在する

最適行動価値関数

- 流れとして、下記のような表(Qテーブル)を作成する
  - 例)状態1において行動3が最も高いQ値であったとき、$\pi_1^*=Q^*(1,3)$となり、状態1においては行動3を取るような方策にする

	行動1	行動2	行動3	行動4
状態1	Q(状態1,行動1)	Q(状態1,行動2)	Q(状態1,行動3)	Q(状態1,行動4)
状態2	Q(状態2,行動1)	Q(状態2,行動2)	Q(状態2,行動3)	Q(状態2,行動4)

最大の時、状態1における方策は「行動3を選択すること」になる

最適行動価値関数

- 最適行動価値関数の導出は省略
- 具体例を示すことで、少し理解できれば良いと思います

5.基本的な解法

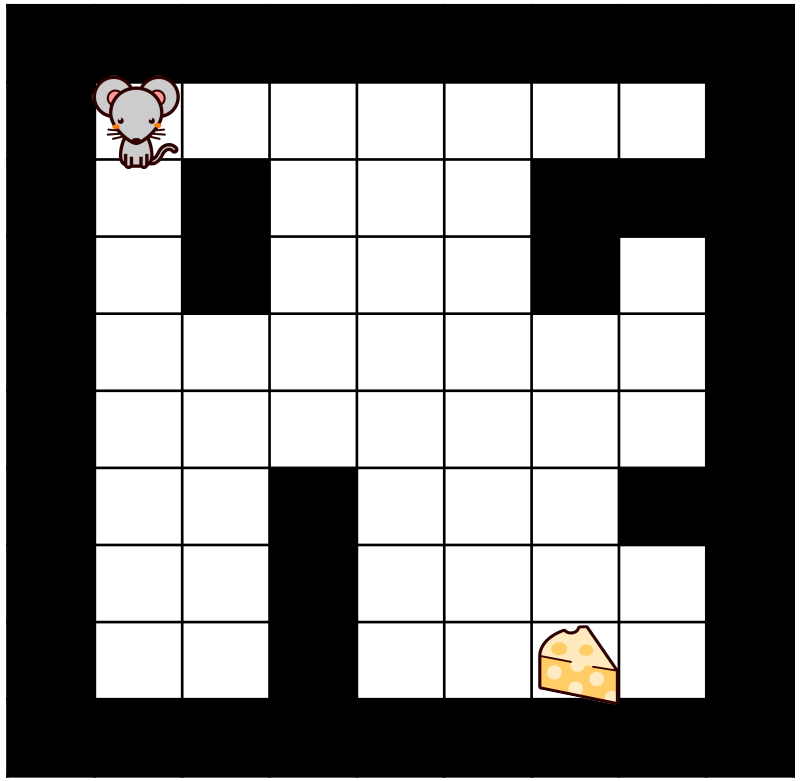
基本的な解法

- 最適な方策を求めるためにはいくつか解法が存在します。
  - 動的計画法
  - モンテカルロ法
  - TD学習
- 今回はTD学習の中のQ学習と呼ばれる方法で、問題を解決したいと思います。

6. 実際に解いてみよう

今回考える問題

Q学習でネズミをチーズまでたどり着けるよう鍛えあげましょう



Q学習とは

以下の式で行動価値関数の更新が行われる

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a} Q(s',a')\right) \cdots (2)$$

s: 状態
a: 行動
$Q(s,a)$: Q値
α:学習率$(0 \leq \alpha < 1)$
γ:割引率$(0 \leq \gamma < 1)$

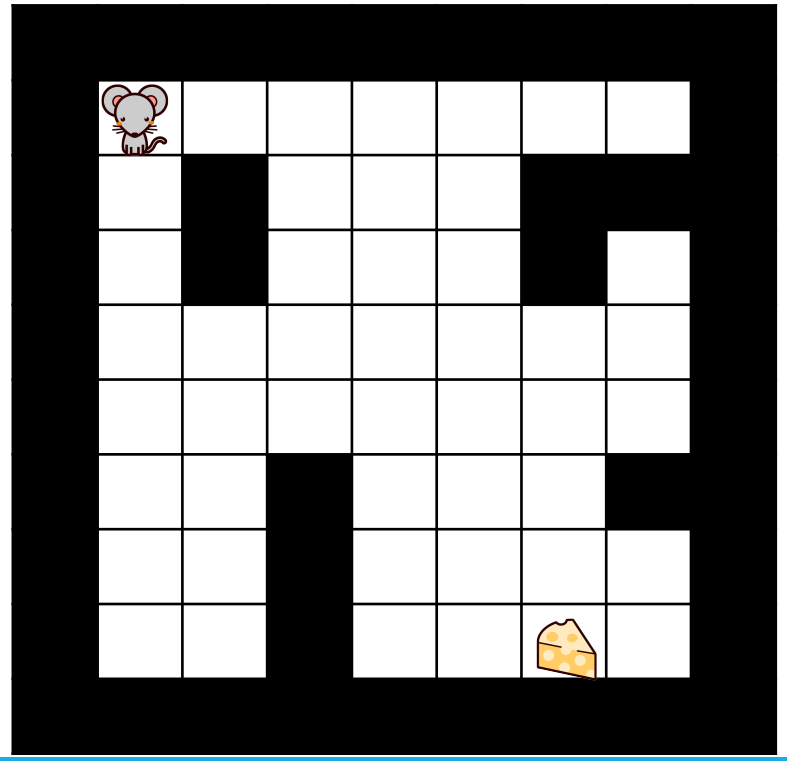
$r(s,a)$: 状態sにおいてaを実行したときに得られる報酬
s': 状態sにおいてaを実行した時の遷移先
a': 遷移先の状態で行う行動

エージェントと環境

- 強化学習では「エージェント」と「環境」を考える必要がある
  - 「エージェント」: ネズミ , 「環境」: 迷路
- 環境である迷路は状態をもつ
  - 状態 : 迷路上それぞれのマス目
- エージェントであるネズミはそれぞれの状態において行動を持つ
  - 行動 : {”左”,”右”,”上”,”下”}
  - どの状態においても、同じ行動が可能

状態

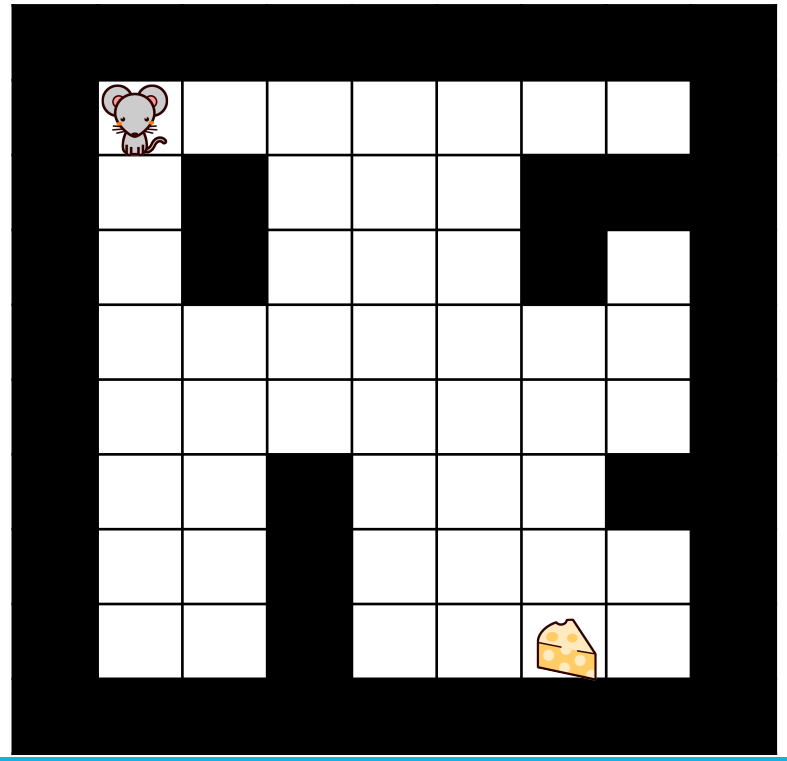
- 迷路のもつ状態sを以下のようにする
  - $s = \{0,1,2 \ldots, 89\}$



0	10	20	30	40	50	60	70	80
1	11	21	31	41	51	61	71	81
2	12	22	32	42	52	62	72	82
3	13	23	33	43	53	63	73	83
4	14	24	34	44	54	64	74	84
5	15	25	35	45	55	65	75	85
6	16	26	36	46	56	66	76	86
7	17	27	37	47	57	67	77	87
8	18	28	38	48	58	68	78	88
9	19	29	39	49	59	69	79	89

報酬

- それぞれの状態における報酬を決める
  - 今回は「壁：−1」,「チーズ:10」,「それ以外の道:0」とする



-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	0	0	0	0	0	0	0	-1
-1	0	-1	0	0	0	-1	-1	-1
-1	0	-1	0	0	0	-1	0	-1
-1	0	0	0	0	0	0	0	-1
-1	0	0	0	0	0	0	0	-1
-1	0	0	-1	0	0	0	0	-1
-1	0	0	-1	0	0	0	0	-1
-1	0	0	-1	0	0	10	0	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

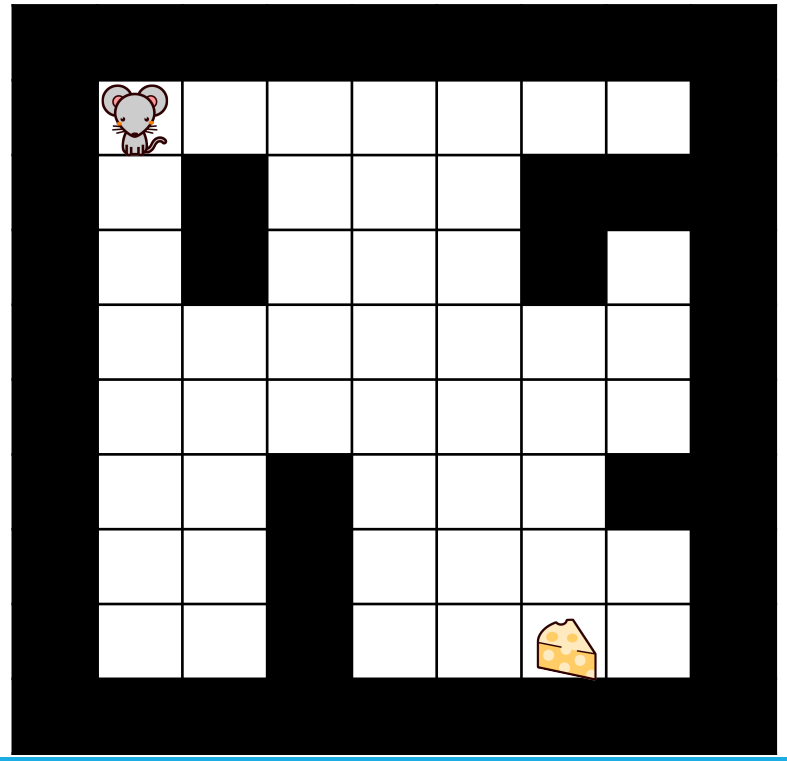
Q値の表を準備

- Q値の表を用意(初期値は0.0)

	行動1(右)	行動2(下)	行動(左)	行動4(上)
状態1	0.0	0.0	0.0	0.0
状態2	0.0	0.0	0.0	0.0
		⋮		
状態88	0.0	0.0	0.0	0.0
状態89	0.0	0.0	0.0	0.0

ネズミの初期位置

- 実際にネズミを動かし、報酬を与えつつ学習させる
  - ネズミの初期位置は状態$s = 11$である



0	10	20	30	40	50	60	70	80
1	11	21	31	41	51	61	71	81
2	12	22	32	42	52	62	72	82
3	13	23	33	43	53	63	73	83
4	14	24	34	44	54	64	74	84
5	15	25	35	45	55	65	75	85
6	16	26	36	46	56	66	76	86
7	17	27	37	47	57	67	77	87
8	18	28	38	48	58	68	78	88
9	19	29	39	49	59	69	79	89

ネズミが探索を行う

- 確率εでネズミの行動を決定
  - ε :ランダムに行動を選ぶ
  - (1-ε) : Q値が最も高くなる行動を選ぶ
  - ちなみに、、、これ「ε-greedy法」
- 以下の更新式において
  - $\alpha = 0.5, \gamma = 0.9$ としておく

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a} Q(s',a')\right) \cdots (2)$$

Q値の更新

- ε：ランダムに行動を選ぶ

状態に対する報酬

-1	-1	-1
-1	0	0
-1	0	-1

状態

0	10	20
1	11	21
2	12	22

現在ネズミがおかれている状態(迷路の上)におけるQ値

	行動1(右)	行動2(下)	行動3(左)	行動4(上)
状態11	0.0	0.0	0.0	0.0

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a} Q(s',a')\right)$$より

最大値が同じ時はどれでもよい

行動aによってネズミが置かれる状態(迷路の上)におけるQ値

	行動1(右)	行動2(下)	行動3(左)	行動4(上)
状態21	0.0	0.0	0.0	0.0

Q値の更新

というわけで、、、

$$Q(11, 右) = (1 - 0.5) * 0.0 + 0.5(0.0 + 0.9 * 0.0) = 0.0$$

となり、更新

	行動1(右)	行動2(下)	行動3(左)	行動4(上)
状態11	0.0	0.0	0.0	0.0

一応更新(試行回数が増えれば値が変わっていく)

Q値の更新

- (1-ε) : Q値が最も高くなる行動を選ぶ時
  - ※Q値が全て同じ時はランダムに選ぶ、、、

状態に対する報酬

-1	-1	-1
-1	0	0
-1	0	-1

状態

0	10	20
1	11	21
2	12	22

現在ネズミがおかれている状態(迷路の上)におけるQ値

	行動1(右)	行動2(下)	行動3(左)	行動4(上)
状態11	0.0	0.0	0.0	0.0

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left(r(s,a) + \gamma \max_{a} Q(s',a')\right)$$より

最大値が同じ時はどれでもよい

行動aによってネズミが置かれる状態(迷路の上)におけるQ値

	行動1(右)	行動2(下)	行動3(左)	行動4(上)
状態1	0.0	0.0	0.0	0.0

Q値の更新

というわけで、、、

$$Q(11, \text{左}) = (1 - 0.5) * 0.0 + 0.5(0.0 + 0.9 * 0.0) = 0.0$$

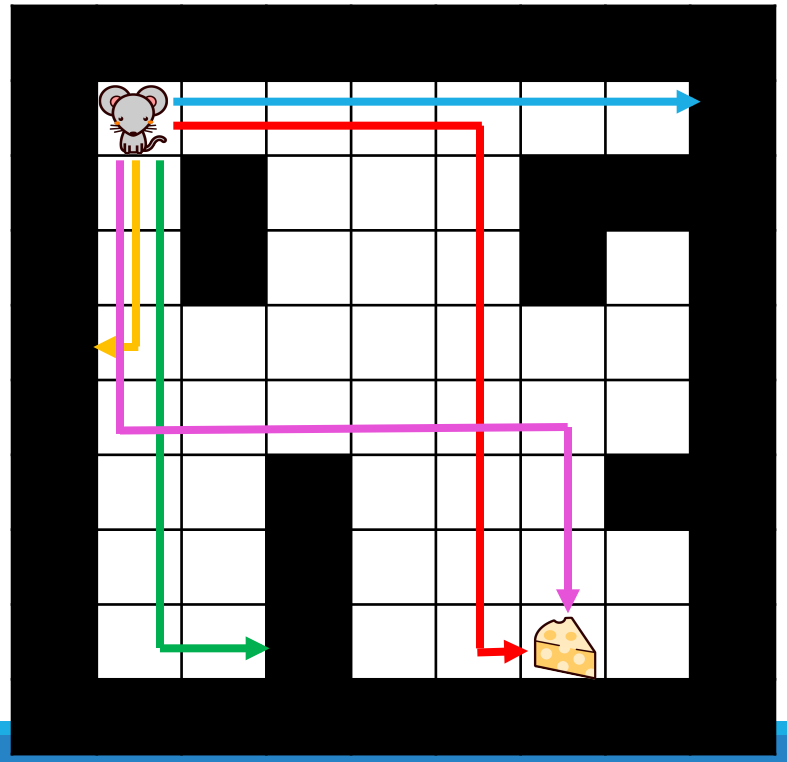
となり、更新

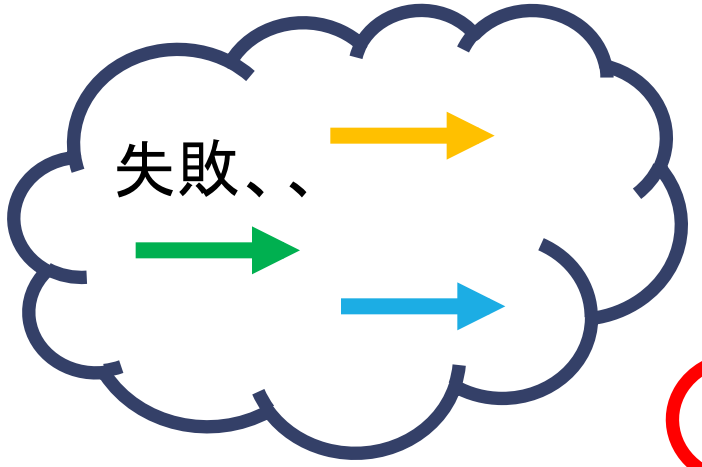
	行動1(右)	行動2(下)	行動3(左)	行動4(上)
状態11	0.0	0.0	0.0	0.0

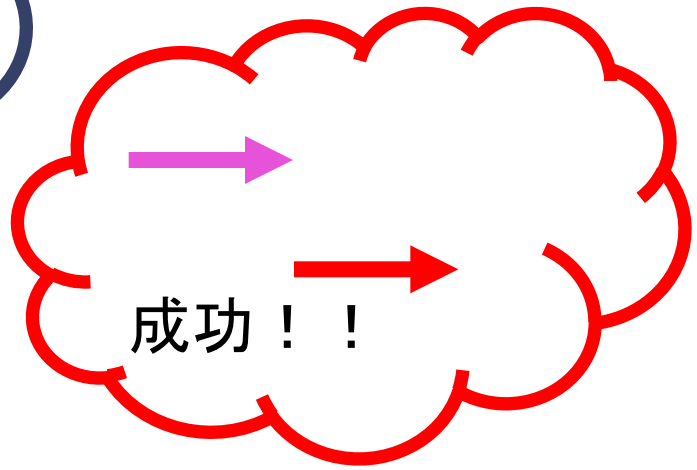
一応更新(試行回数が増えれば値が変わっていく)

ネズミを鍛えあげる

- 多くの試行を行うことでネズミは学習していく
- 「失敗」か「成功」をした時に、ネズミを初期位置に戻し試行再開







ネズミを鍛えあげる

試行が全て終わった後のQ値の表

	行動1(右)	行動2(下)	行動(左)	行動4(上)
状態1	0.0	0.0	0.0	0.0
		⋮		
状態11	3.13	0.0	-0.99	-0.99
		⋮		
状態21	3.48	-0.99	2.73	-0.99
		⋮		
状態88	0.0	0.0	0.0	0.0
状態89	0.0	0.0	0.0	0.0

Q値の表にしたがって行動選択

それぞれの状態において、Q値が最大になる行動を選択

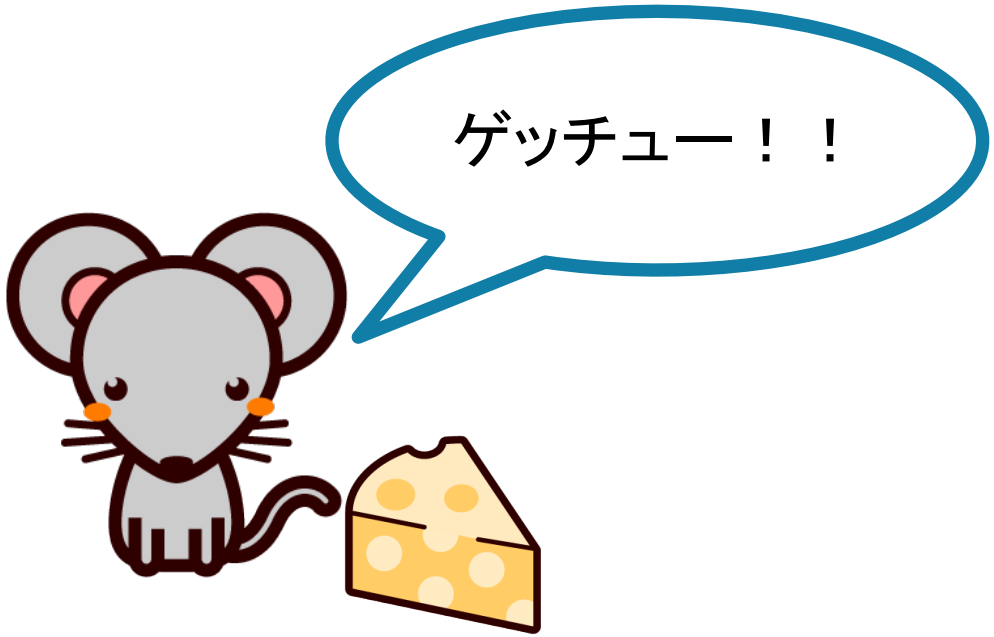
0	10	20	30	40	50	60	70	80
1	11	21	31	41	51	61	71	81
2	12	22	32	42	52	62	72	82
3	13	23	33	43	53	63	73	83
4	14	24	34	44	54	64	74	84
5	15	25	35	45	55	65	75	85
6	16	26	36	46	56	66	76	86
7	17	27	37	47	57	67	77	87
8	18	28	38	48	58	68	78	88
9	19	29	39	49	59	69	79	89

	行動1(右)	行動2(下)	行動(左)	行動4(上)
状態1	0.0	0.0	0.0	0.0
⋮				
状態11	3.13	0.0	-0.99	-0.99
⋮				
状態21	3.48	-0.99	2.73	-0.99
⋮				

結果

- すべての状態においてQ値が最大な行動を選択する
  - そうするとチーズへたどり着けるように学習される

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	→	→	↓	←	←	0	0	-1
-1	0	-1	↓	↑	0	-1	-1	-1
-1	0	-1	→	↓	↓	-1	0	-1
-1	0	0	↑	↓	←	0	0	-1
-1	→	→	→	↓	↓	←	0	-1
-1	0	0	-1	→	→	↓	-1	-1
-1	0	0	-1	↑	→	↓	0	-1
-1	0	0	-1	0	0	10	←	-1
-1	1	-1	-1	-1	-1	-1	-1	-1



ご清聴ありがとうございました。