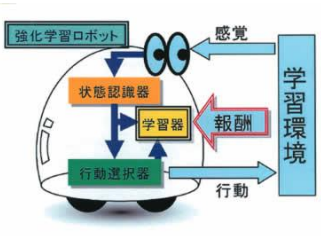
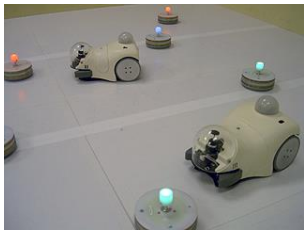
【強化学習】

- 試行錯誤をくりかえして、よりよい行動方針を獲得する手法
- 状態と行動をセットにして記述し、うまくいった場合に「報酬」、失敗した場合に「罰」を与えることでよりよい行動を獲得するようになる

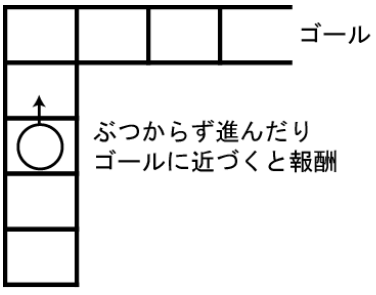


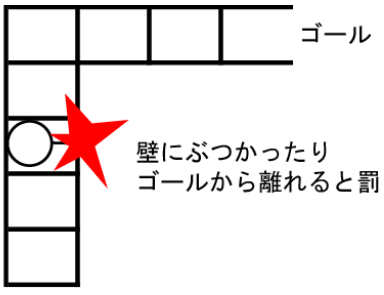


- 強化学習では学習をおこなう「主体」と「環境」がある
- 主体は環境の状態を観測し、行動を選択する
- 行動選択の結果として、環境から「報酬」または「罰」を得る（報酬は毎回与えられるとは限らず、特定の状況でのみ与えられる場合もある）

【単純な強化学習モデル】

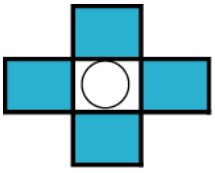
- 壁に囲まれた通路を歩いて、ゴールを目指すモデル
- 行動する主体（エージェント）の行動について以下のように仮定する
  - **上下左右の１マス分を観察できる**
  - **１回につき1マス移動できる**
  - **無事に進めたら報酬、壁にぶつかったら罰を与えられる**
  - **ゴールに近づいたら報酬、ゴールから離れたら罰を与えられる**





【主体（エージェント）の環境】

- エージェントにとっての環境は「観察できる範囲に壁があるか、ないか」
- 観察できる範囲は上下左右の４マス
- 移動した位置がゴールに近づいたか離れたかを知ることが出来る

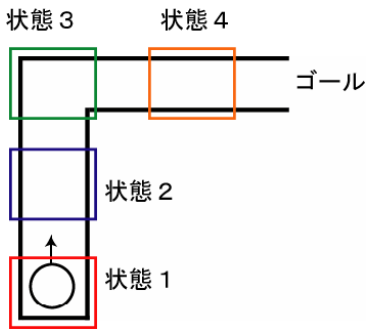


観察範囲

【主体の状況】

■ 先ほどの通路の例では、エージェントが観察できる状況は以下の4種類

- 左右と下が壁、上は空き　（状態1）
- 左右が壁、上下は空き　（状態2）
- 左と上が壁、右と下は空き　（状態3）
- 上下が壁、左右は空き　（状態4）



■ 簡単にするためエージェントは常に上を向いていると仮定する

■ 行動の評価値はすべて5から開始する

■ ゴールに到達したら終了

【報酬の与え方と行動選択】

■ エージェントが壁にぶつからずに進んだら＋1、さらにゴールに近づいたら＋1、ゴールから離れたら－1、壁にぶつかったら－1を評価値に加える

■ 行動選択は「その状況において最も評価値の高いもの」を選ぶこととし、同じ評価値のものが複数ある場合はランダムに1つを選ぶ （greedy 法）

【実行例】

（1） スタート地点＝状態1

全ての行動の評価値は5で同じなので、ランダムに行動選択（「右」が実行されたとする）

壁にぶつかったので（状態1－右）の評価値を－1

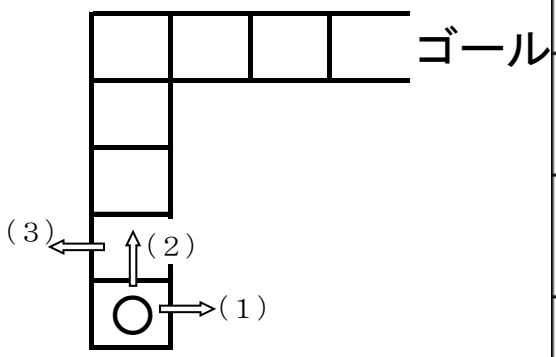
（2） 位置はかわらなかったので（状態1）、上、下、左の評価値が5で最大なのでランダムに選択（「上」が実行されたとする）

ぶつからず、かつゴールに近づいたので（状態1－上）の評価値を＋2

（3） 1マス進んで（状態2）になったため、行動の評価値はすべて5

ランダムに行動を選択する（「左」が実行）

壁にぶつかったので（状態2－左）の評価値を－1、位置はかわらず



		（0）	（1）	（2）	（3）	（4）
状態	行動	評価値	評価値	評価値	評価値	評価値
1	上	5	5	7	7	
	下	5	5	5	5	
	左	5	5	5	5	
	右	5	4	4	4	
2	上	5	5	5	5	
	下	5	5	5	5	
	左	5	5	5	4	
	右	5	5	5	5	
3	上	5	5	5	5	
	下	5	5	5	5	
	左	5	5	5	5	
	右	5	5	5	5	
4	上	5	5	5	5	
	下	5	5	5	5	
	左	5	5	5	5	
	右	5	5	5	5	

行動系列：スタート→右→上→左→・・・

※ゴールに到達、もしくは15回行動選択をおこなったら終了して次回へ

		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
状態	行動	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値
1	上	5															
	下	5															
	左	5															
	右	5															
2	上	5															
	下	5															
	左	5															
	右	5															
3	上	5															
	下	5															
	左	5															
	右	5															
4	上	5															
	下	5															
	左	5															
	右	5															

１回目の行動系列：スタート→

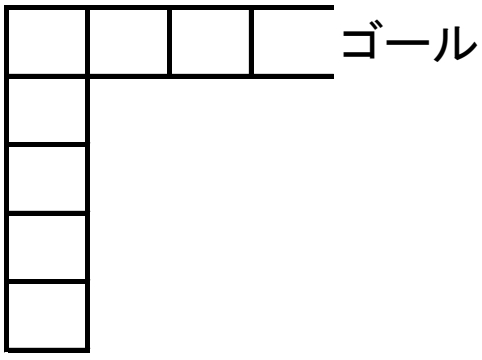
1回目終了時の評価値を入れる

↓

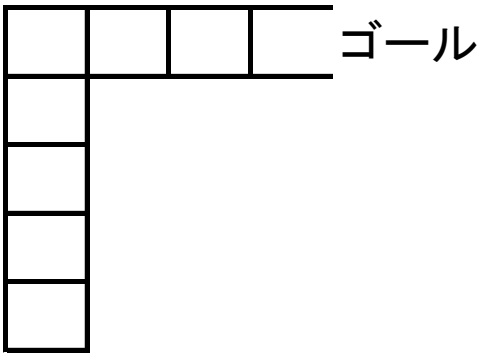
		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
状態	行動	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値	評価値
1	上																
	下																
	左																
	右																
2	上																
	下																
	左																
	右																
3	上																
	下																
	左																
	右																
4	上																
	下																
	左																
	右																

２回目の行動系列：スタート→

1回目



2回目



乱数表（1～4）

4	4	1	1	4	3	2	3	1	1
1	4	4	2	2	4	4	4	2	1
3	4	3	2	1	1	1	3	3	4
1	1	1	4	3	4	2	4	4	4
2	3	4	1	4	2	1	1	4	4
3	4	4	1	1	4	1	4	4	2
2	1	2	2	2	3	4	3	4	3
2	3	3	1	2	3	2	4	3	2
1	4	3	3	1	2	1	1	4	2
1	2	1	1	2	3	2	1	2	3

乱数表（1～3）

3	2	3	3	2	1	2	2	3	1
2	2	1	3	1	1	2	3	2	2
1	2	3	1	3	3	3	2	1	1
3	2	3	3	3	3	3	3	1	3
1	3	3	2	3	3	1	1	3	1
2	3	3	1	1	3	3	3	1	1
2	1	1	3	2	2	3	3	2	2
1	1	3	2	2	2	3	3	3	1
2	1	3	3	2	3	2	1	3	1
1	1	1	2	2	3	2	2	2	2

乱数表（1～2）

2	1	1	2	2	2	1	1	1	2
2	2	2	2	1	1	1	1	2	1
1	2	1	1	2	1	1	2	2	1
1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	2	2	1	1
2	1	1	1	1	2	1	1	1	2
2	1	2	2	2	2	1	2	2	2
1	1	2	2	2	2	1	1	1	2
1	1	1	2	1	2	1	2	1	2
1	2	2	2	2	2	1	2	2	2

【Q-leaning】

- 強化学習の代表的アルゴリズム
- Q 値と呼ばれる「環境と行動の組み合わせ」の評価値を逐次修正してゆき、最適な行動を探す方法
- Q-learning は行動により状態が変わった後の「仮定の行動」を用いて評価をおこなう→Off-Policy の方式
- これに対し、On-Policy と呼ばれるものは厳密に「自分が行動した結果」に基づいて評価をおこなう
  - 代表的手法として profit sharing など（報酬を得た時点から過去の行動にさかのぼって報酬を与える方式）
- 強化学習には様々な方式があり、それぞれに特徴がある
- 状況や問題に応じて使い分ける

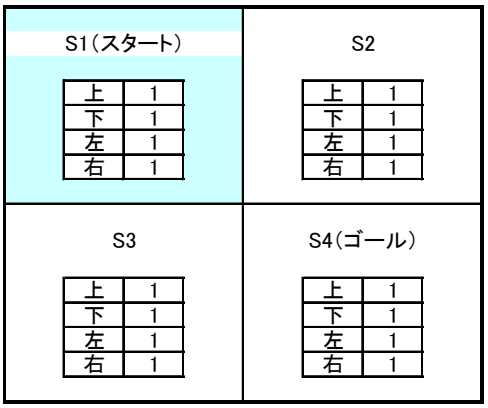
（1） エージェントは環境の状態 s_t を観測する．
（2） エージェントは任意の行動選択方法 (探査戦略) に従って行動 a_t を実行する．
（3） 環境から報酬 r_t を受け取る．
（4） 状態遷移後の状態 s_{t+1} を観測する．
（5） 以下の更新式により Q 値を更新：

$$Q(s_t, a_t) \leftarrow (1-\alpha)\, Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_a Q(s_{t+1}, a) \right]$$

ただし α は学習率 $(0 < \alpha \leq 1)$，γ は割引率 $(0 \leq \gamma < 1)$ である．
（6） 時間ステップ t を $t+1$ へ進めて手順 1 へ戻る．

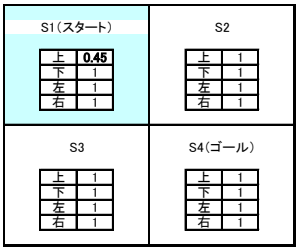
【数値例】

- 例）以下のような4マスの迷路を考える
- 各マスでの状態をそれぞれ S1～S4 とし、行動は上下左右の4種をとることができるものとする
- マスの一番外の枠は壁とし、壁方向へは移動できない（もとの場所にとどまる）
- 壁にぶつかったら報酬 －1、ゴールしたら +1、それ以外は報酬 0 とする
- 学習率 α ＝0.5、割引率 γ ＝0.9 とする



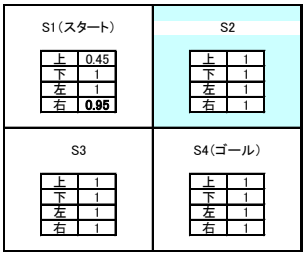
■ S1 からスタートし、行動「上」が選ばれたとすると
→壁に当たるため位置は S1 のまま、報酬は－1

$$Q(S1,\text{上}) \leftarrow (1-0.5)Q(S1,\text{上}) + 0.5[-1 + 0.9 \max_a Q(S1,a)]$$
$$= 0.5\times 1 + 0.5\times(-1+0.9)$$
$$= 0.45$$



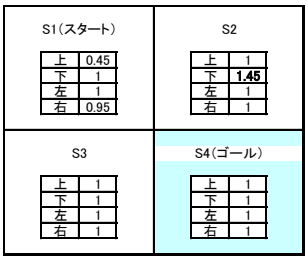
■ 次に、S1 で行動「右」が選ばれたとすると
→状態は S2 へ移動、報酬は 0

$$Q(S1,\text{右}) \leftarrow (1-0.5)Q(S1,\text{右}) + 0.5[0 + 0.9 \max_a Q(S2,a)]$$
$$= 0.5\times 1 + 0.5\times(0.9\times 1)$$
$$= 0.95$$



■ 次に、S2 で行動「下」が選ばれたとすると
→状態は S4（ゴール）へ移動、報酬は 1

$$Q(S2,\text{下}) \leftarrow (1-0.5)Q(S2,\text{下}) + 0.5[1 + 0.9 \max_a Q(S4,a)]$$
$$= 0.5\times 1 + 0.5\times(1+0.9\times 1)$$
$$= 1.45$$



【行動選択】

- Q 値から行動を決定する方法には以下のようなものがある
- **ε -greedy**： ε の確率でランダム、それ以外は最大の重みを持つルールを選択
- **ルーレット選択**： Q(s,a)に比例した割合で行動選択
- **ボルツマン選択**： exp(Q(s,a)/T)に比例した割合で行動選択、ただし T は時間とともに 0 に近付く
  ただし s は環境の状態、a は行動

【レポート】

- 上の数値例と同じ条件で S1 からスタートし、「　　」→「　　」→「　　」→「　　」の順に行動が選択された場合、最終的にＳ１～Ｓ４の Q 値がどうなっているか計算せよ。ただし Q 値の初期値はすべて１とする。