強化学習： Q 学習

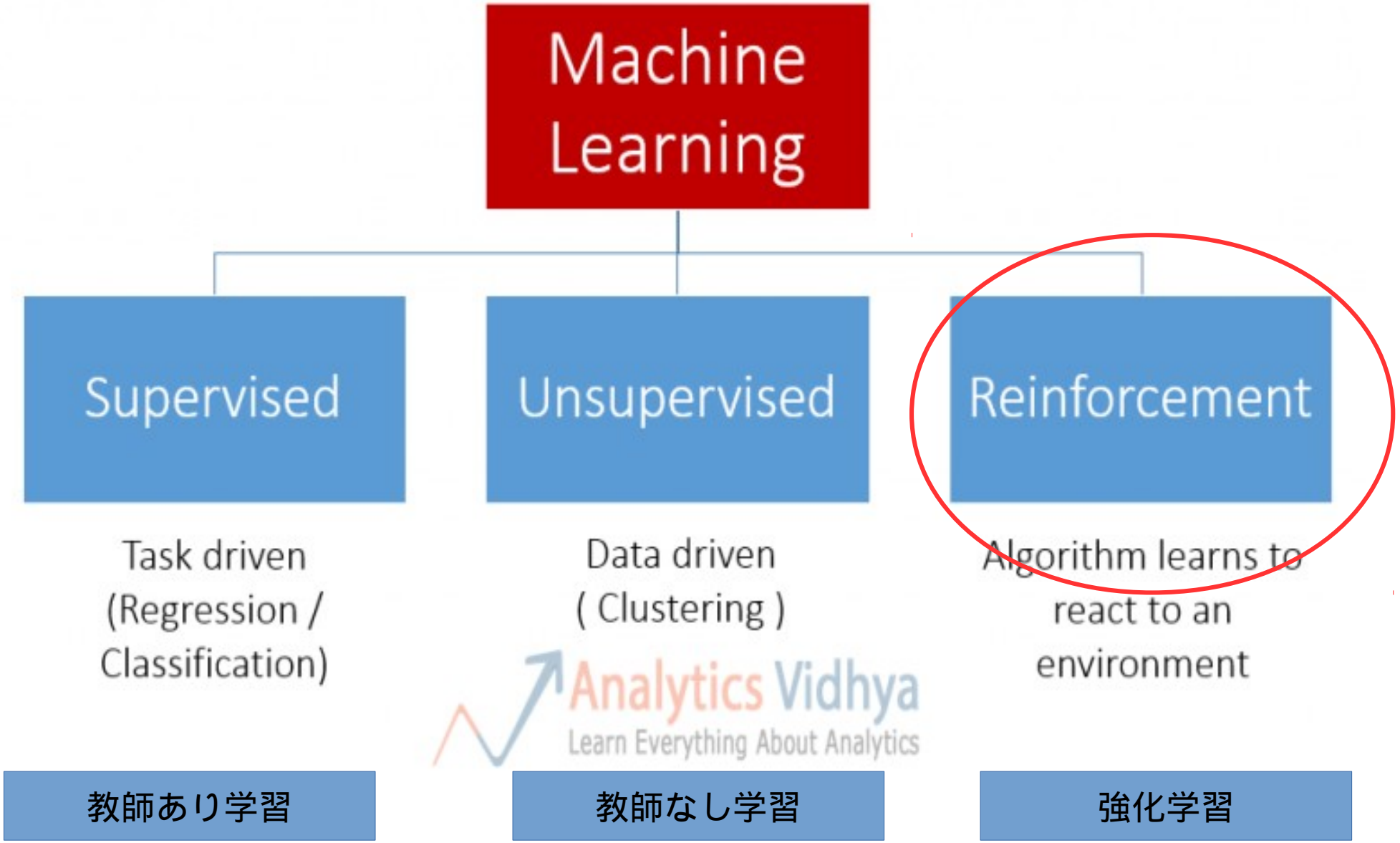
2017/03/23

1426071 スプラトマン ジョシュア

学習とは？

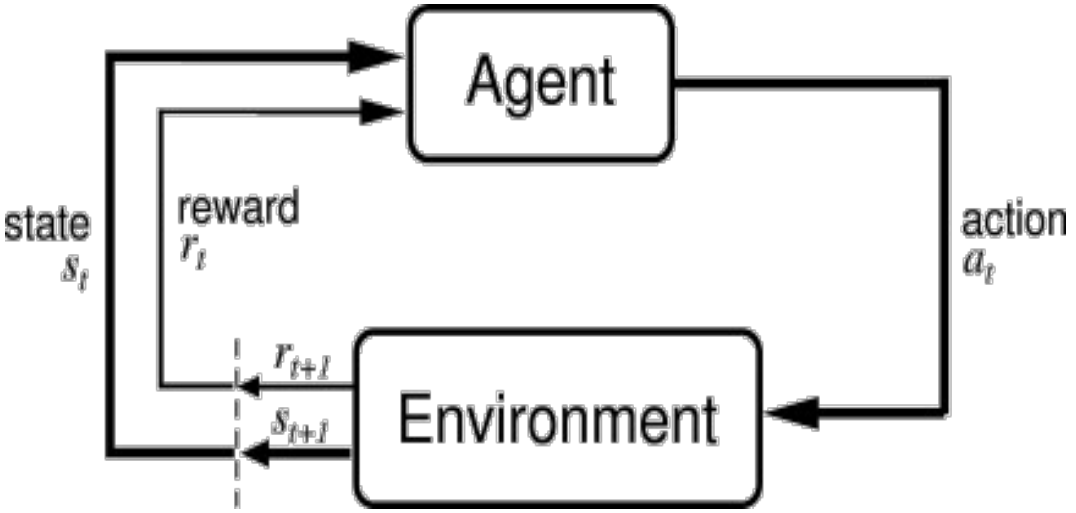
- 人工知能の一部
- 人間の学習能力を機械（コンピュータ）で真似する
- あるデータに対して学習し、そこからパターンを見つけ出す
- https://youtu.be/qv6UVOQ

Types of Machine Learning



強化学習 (RL) とは？

- 状態に対してどんな行動をする学習タイプ
- 最大の報酬を探す
- 赤ちゃんが歩きたい時考えよう！



強化学習 vs

教師あり学習 (SL)

- 教師データを比較しながら学習
- チェスを考える
  - SL: あやゆる戦略を事前対策する
  - RL: 状態に対して行動を決める

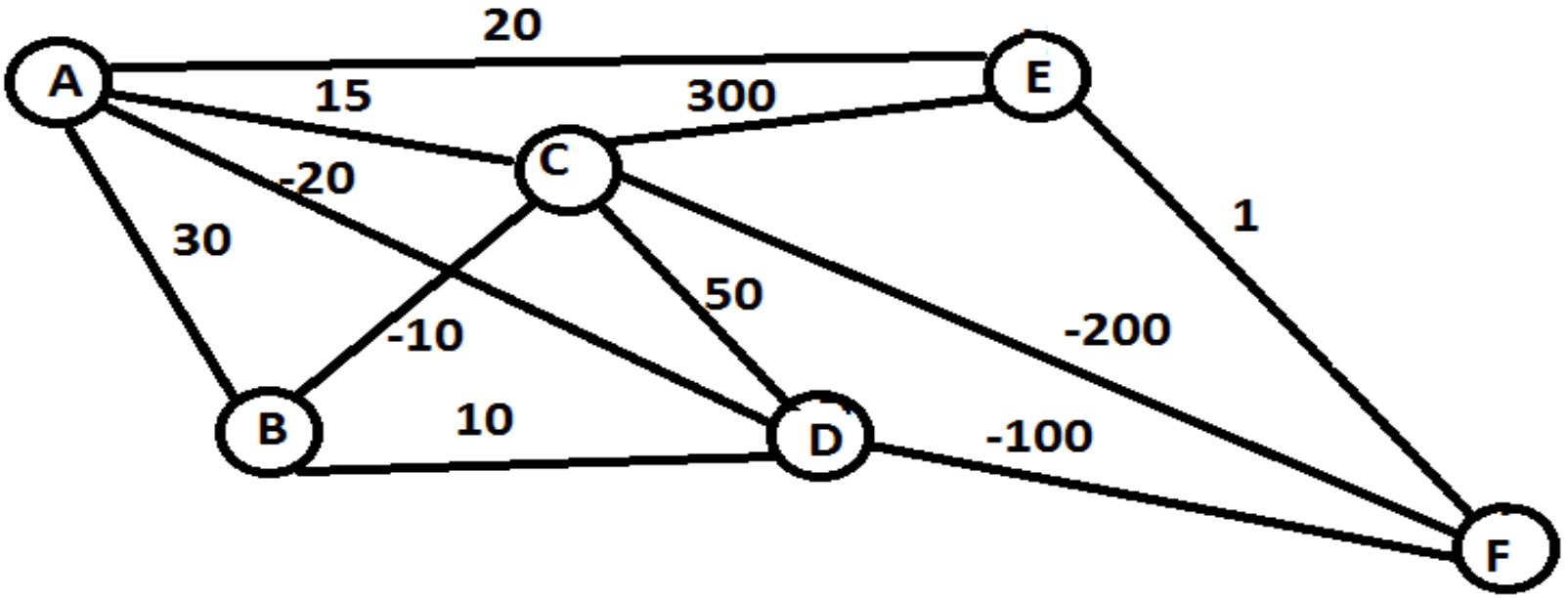
- 教師なし学習 (UL)
- データのパターンを探す
- 好きな雑誌を進める AI を考える
  - UL: 過去読んだ似たような雑誌をお進め
  - RL: たまたま新しい雑誌とか出し、フィードバックを受けながら好みを探す

Basic of 強化学習

- 搾取 vs 探検どっち？適当はダメ
- markov モデル：
  - 状態 :S
  - 行動 :A
  - 報酬 :R
  - ポリシー :π (状態 + 行動)
  - 値 :V

$$E(r_t \mid \pi, s_t)$$

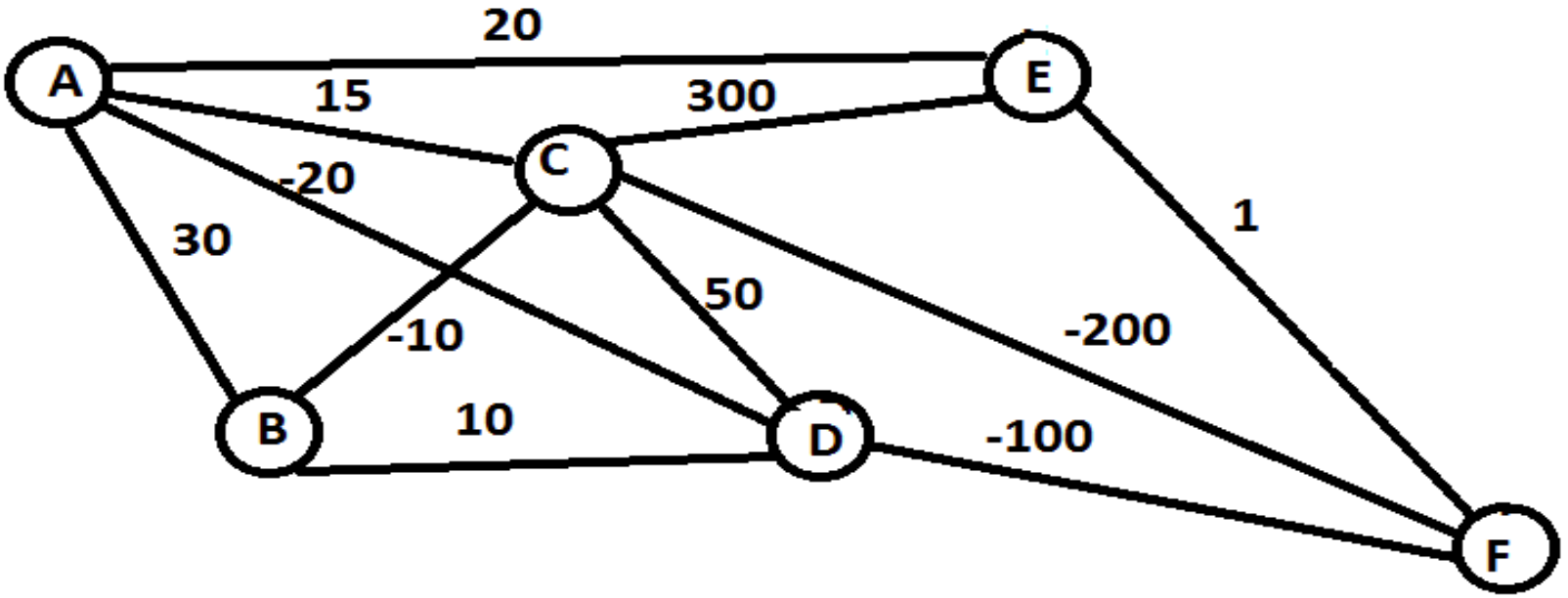
Basic of 強化学習



状態：？　　報酬：？　　値：

行動：？　　ポリシ：？

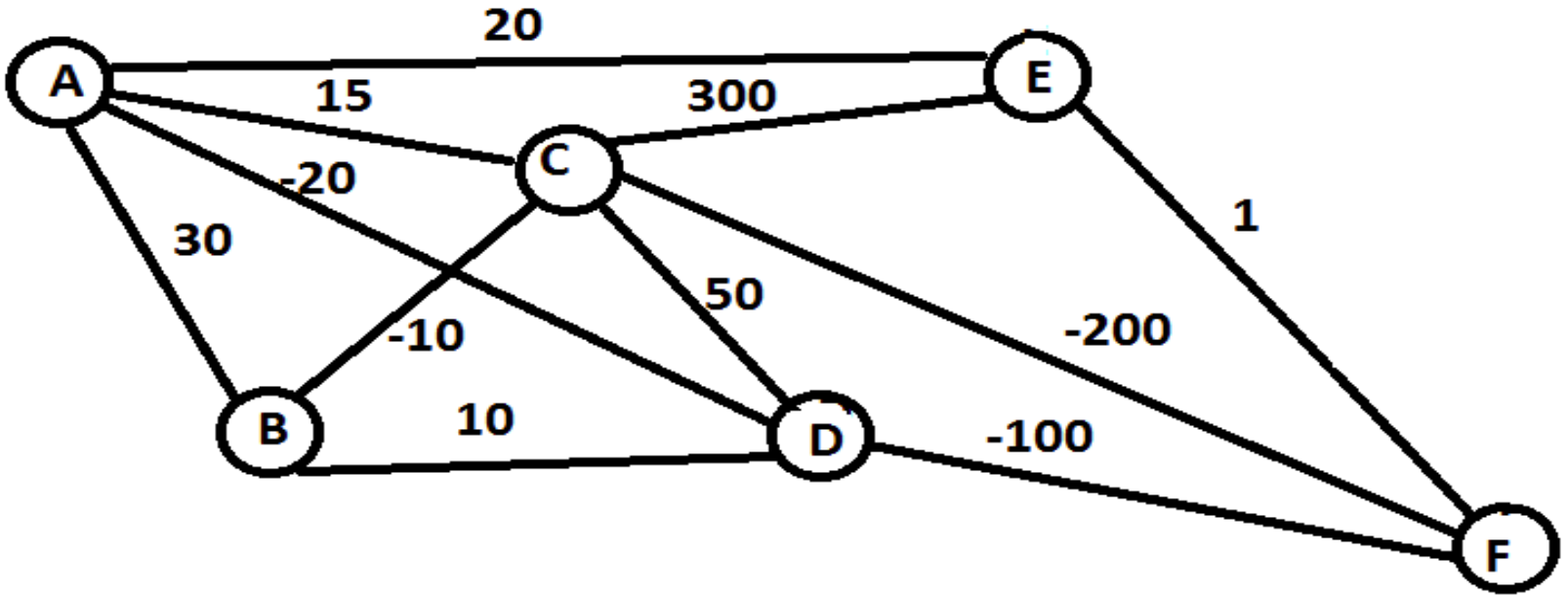
Basic of 強化学習



状態： A,C,D　報酬： 20,15　　値： 20+1,etc

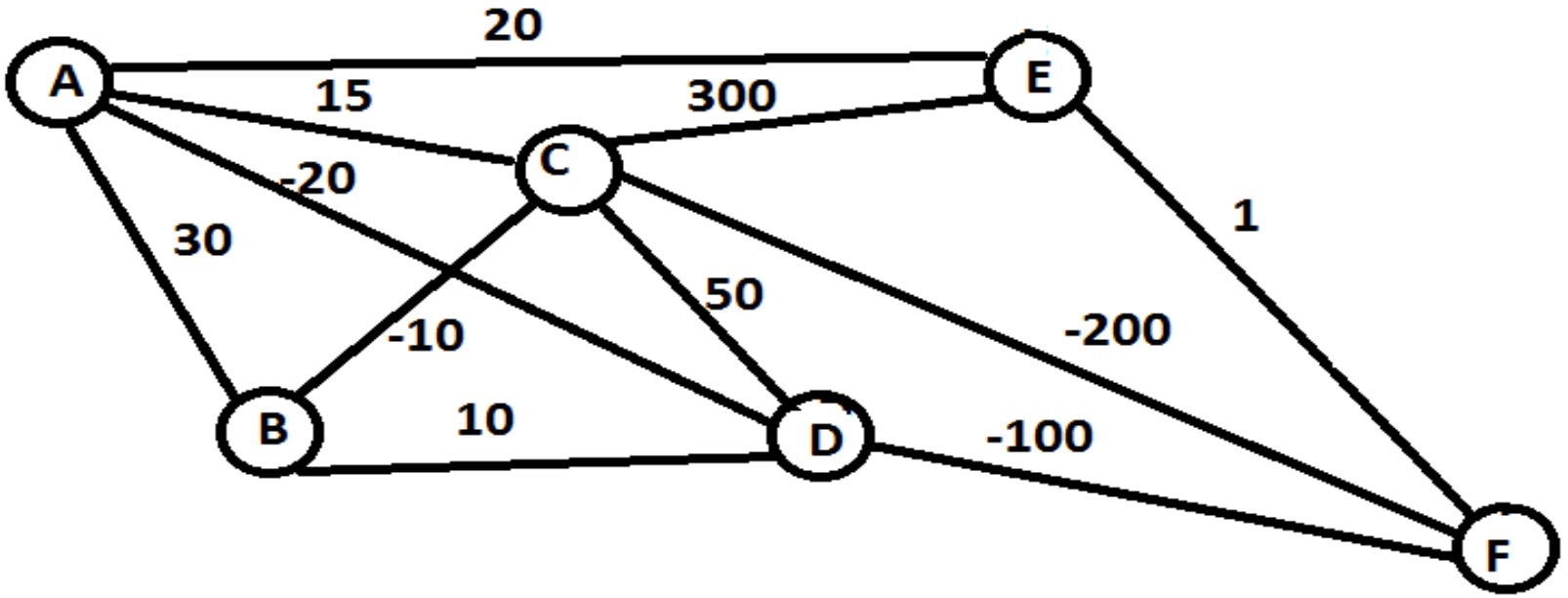
行動： A->C　ポリシ： A-E-F, C-B, C-D-F

Basic of 強化学習



- スタート： A　ゴール： F　最短経路？

Basic of 強化学習



- スタート： A　ゴール： F　最短経路： A-D-F
- A->(B,C,D,E)　A-D が一番コスト低い → -20
- D->(B,C,F)　D-F が一番コスト低い → 値 :-120

おめでとう！

- さっき強化学習アルゴリズムをやりました！
- epsilon greedy 法
  - 搾取 vs 探検どっち？

他のやりかた？

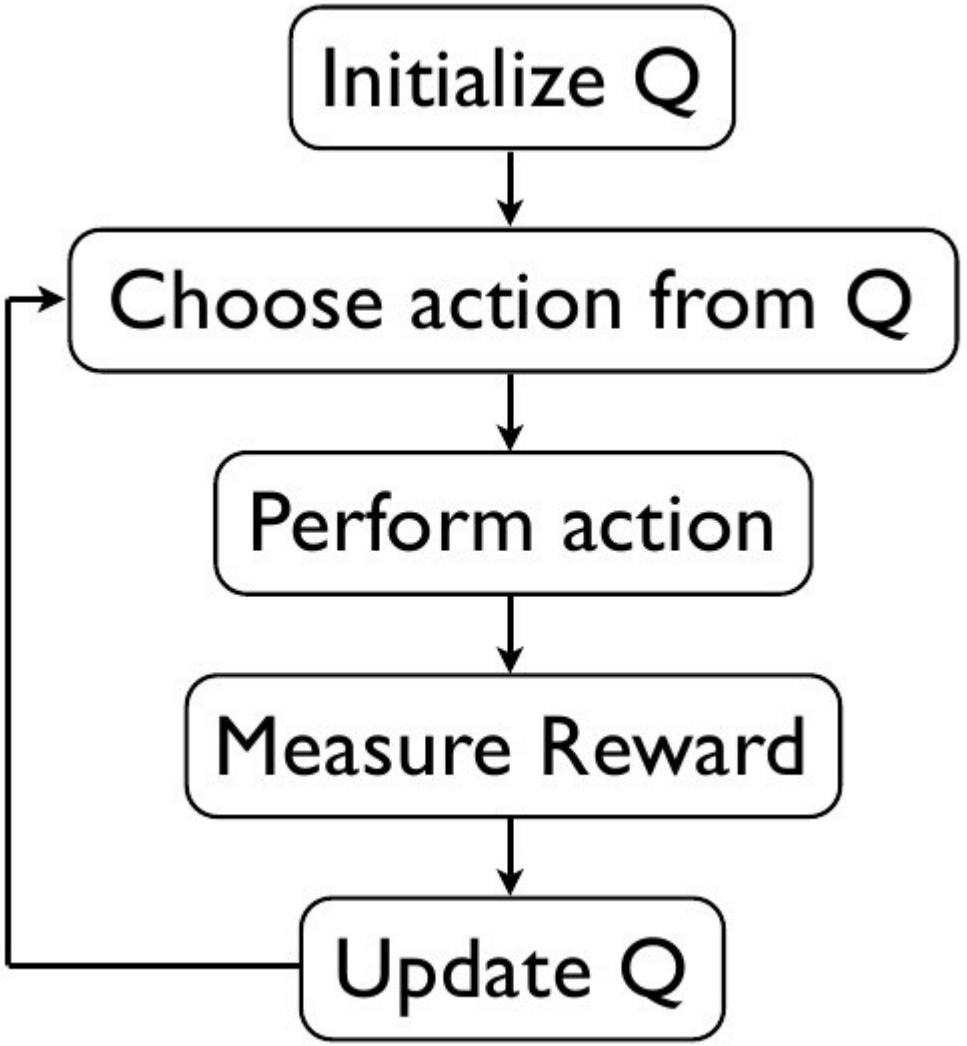
- Value Based　最適の値を重心として進む
- Policy Based　最適のポリシを重心として進む

- ちなみに epsilon greedy 法は？

Q 学習とは？

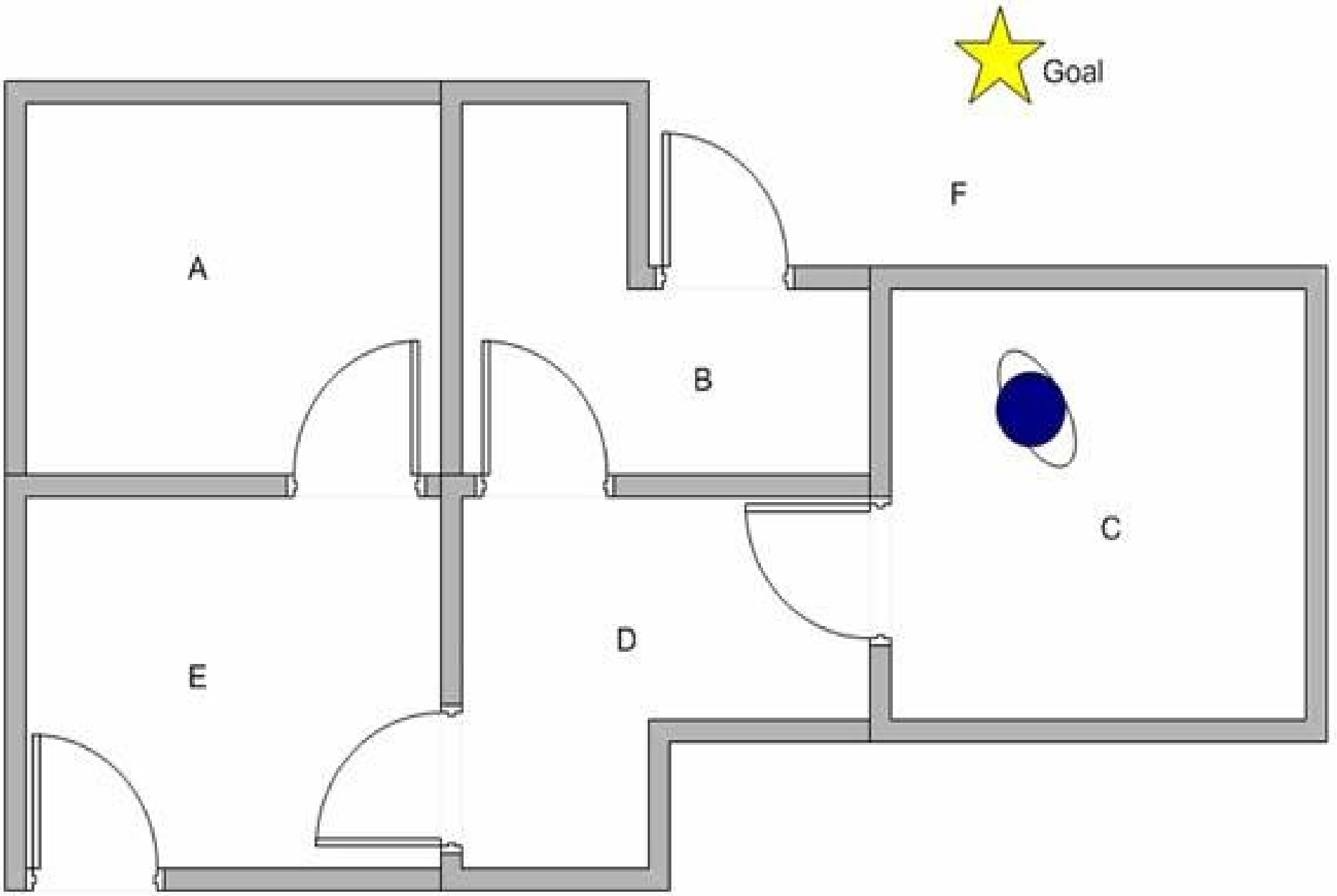
- 強化学習のアルゴリズムの一つ
- Policy Based
- https://youtu.be/JS1fImajAEE
- https://youtu.be/OisMuSW2F2w
- AlphaGo! Neural Network で Q 学習！

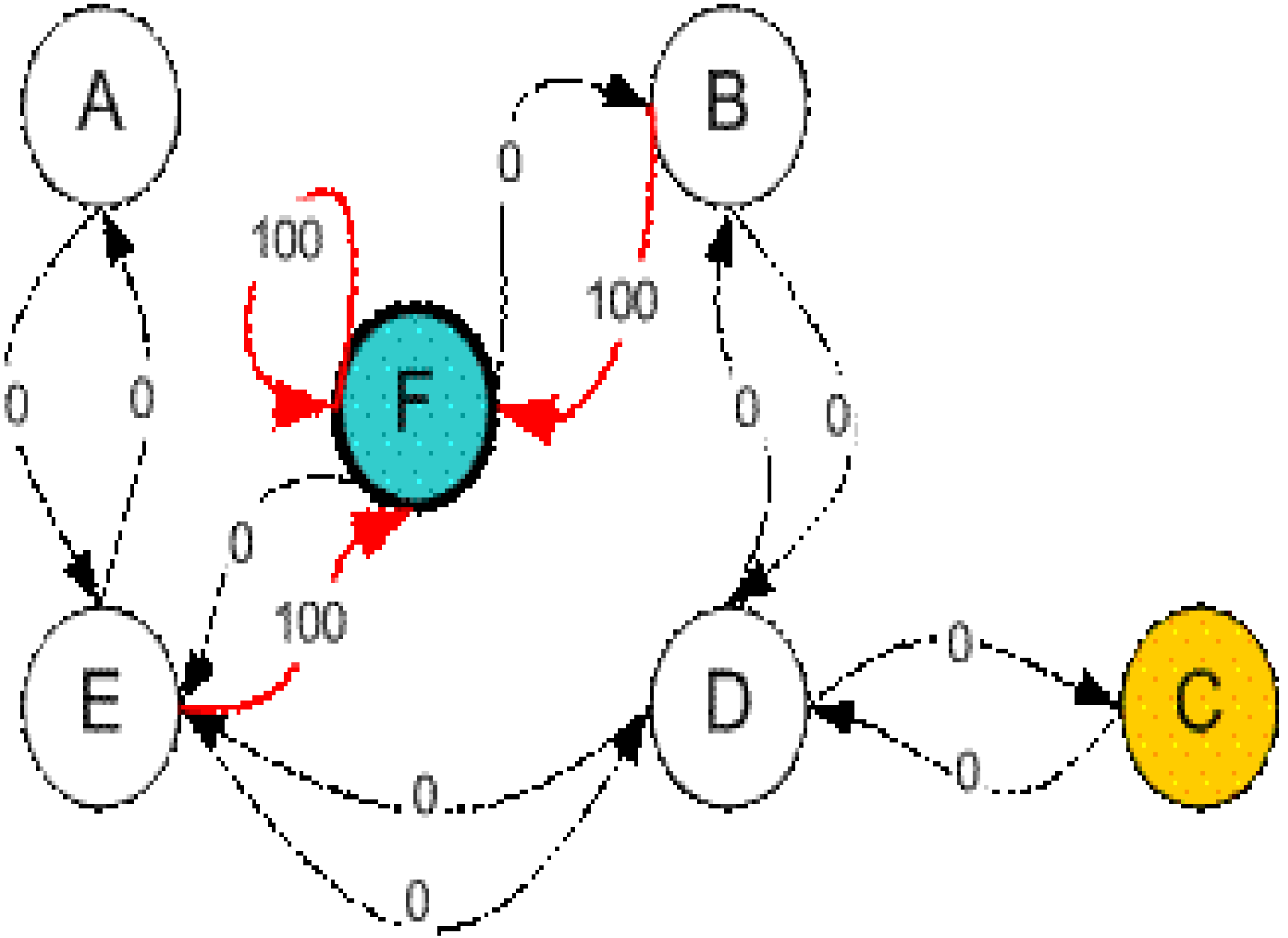
Q 学習のながれ (訓練)


Initialize Q
Choose action from Q
Perform action
Measure Reward
Update Q

Q 学習のながれ

- 今回はシンプルな物を作ります
- Neural Network ではなく QTable (nxn) 行列を使います
- QTable を学習することが目的
- 学習した QTable を使って実装する


Goal
F
A
B
C
D
E

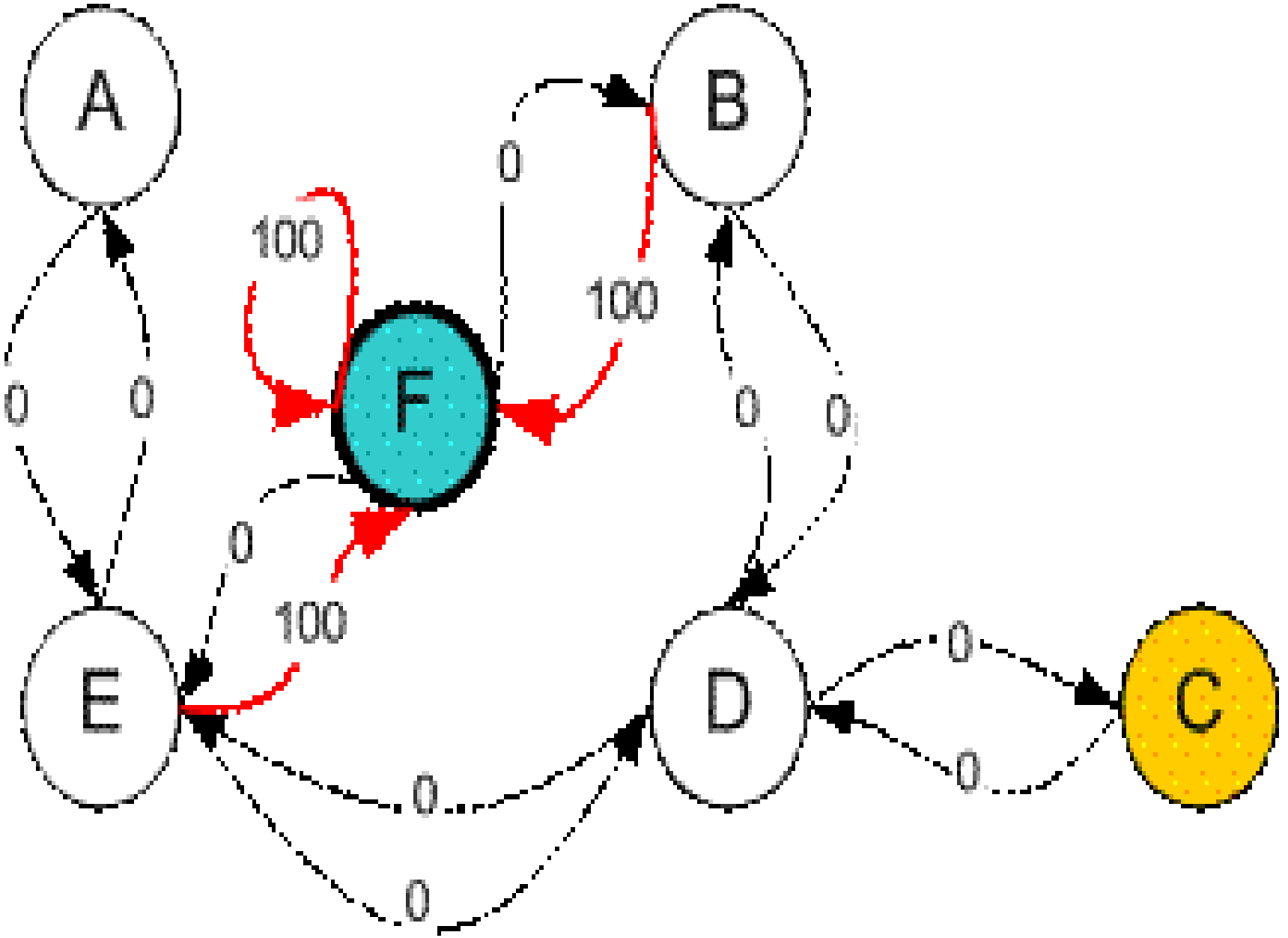

A
B
C
D
E
F
0
100
100
0
0
0
0
0
100
0
0
0
0
0

markov モデル

- 状態 S: ?
- 行動 A: ?
- 報酬 R: ?
- ポリシ π: ?
- 値 V: ?

markov モデル

- 状態 S: A,B,C,D,E,F
- 行動 A: A-E, B-D, B-F, C-D, D-E, E-F, F-F
- 報酬 R: 0,100 <- 行列で表す
- ポリシ π: QTable <- これを学習する
- 値 V: QTable からの値


A
B
C
D
E
F
0
100
100
0
0
0
0
0
100
0
0
0
0
0

QTable と R 行列

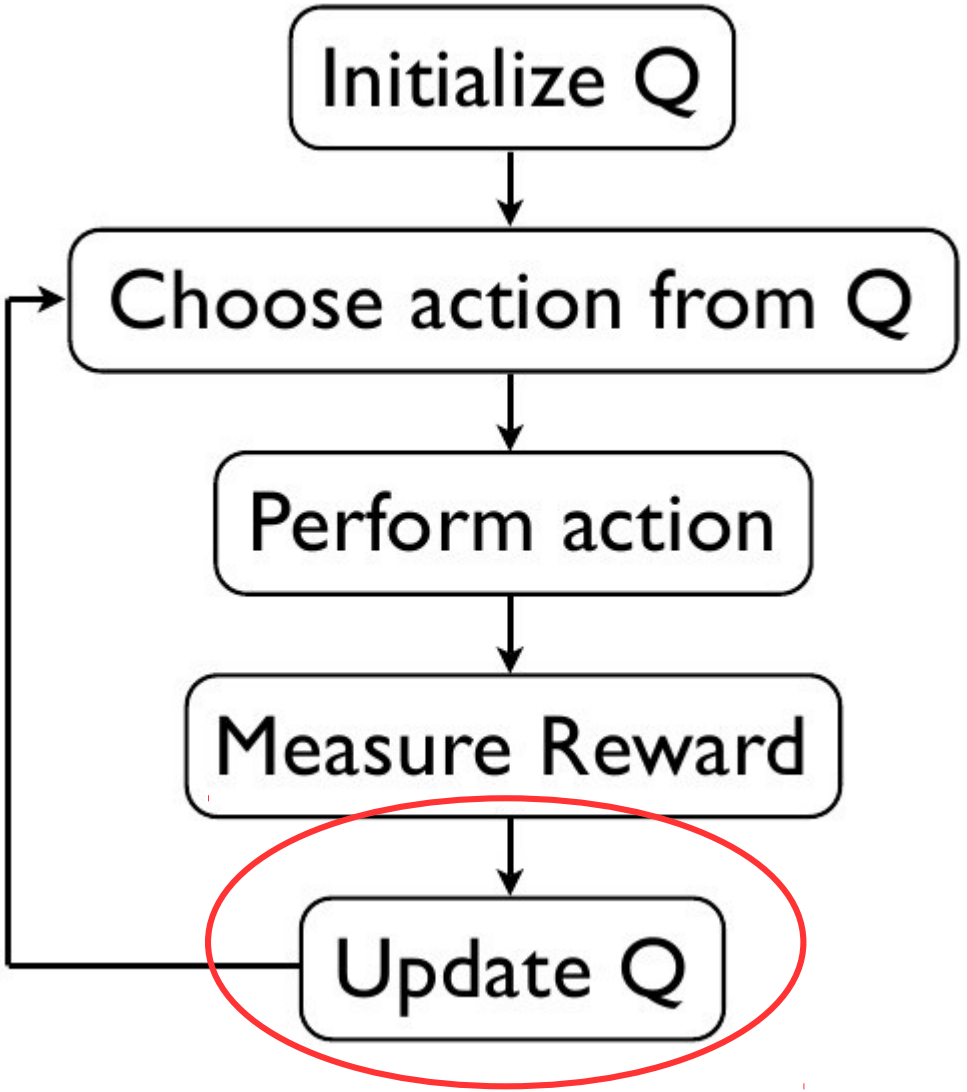
$$\mathbf{R} = \begin{array}{c} \textit{state} \backslash \textit{action} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} - & - & - & - & 0 & - \\ - & - & - & 0 & - & 100 \\ - & - & - & 0 & - & - \\ - & 0 & 0 & - & 0 & - \\ 0 & - & - & 0 & - & 100 \\ - & 0 & - & - & 0 & 100 \end{bmatrix} \end{array}$$

$$\mathbf{Q} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Update 式

$$\mathbf{Q}(state, action) = \mathbf{R}(state, action) + \gamma \cdot Max\left[\mathbf{Q}(next\ state, all\ actions)\right]$$

- y は学習パラメータ (0~1)
- 報酬がすぐか遅れるか



やってみよう！

$$\mathbf{Q} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

- B にいる場合
- ランダムに F に移動
- セッション終了

$$\mathbf{Q}(state, action) = \mathbf{R}(state, action) + \gamma \cdot Max\left[\mathbf{Q}(next\ state, all\ actions)\right]$$

$$\mathbf{Q}(B,F) = \mathbf{R}(B,F) + 0.8 \cdot Max\left\{\mathbf{Q}(F,B), \mathbf{Q}(F,E), \mathbf{Q}(F,F)\right\} = 100 + 0.8 \cdot 0 = 100$$

$$\mathbf{Q} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

やってみよう！

$$\mathbf{Q} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

- D にいる場合
- ランダムに B に移動
- ゴールまで続く

$$\mathbf{Q}(state, action) = \mathbf{R}(state, action) + \gamma \cdot Max\left[\mathbf{Q}(next\ state, all\ actions)\right]$$

$$\mathbf{Q}(D,B) = \mathbf{R}(D,B) + 0.8 \cdot Max\left\{\mathbf{Q}(B,D), \mathbf{Q}(B,F)\right\} = 0 + 0.8 \cdot Max\left\{0, 100\right\} = 80$$

$$\mathbf{Q} = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 80 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

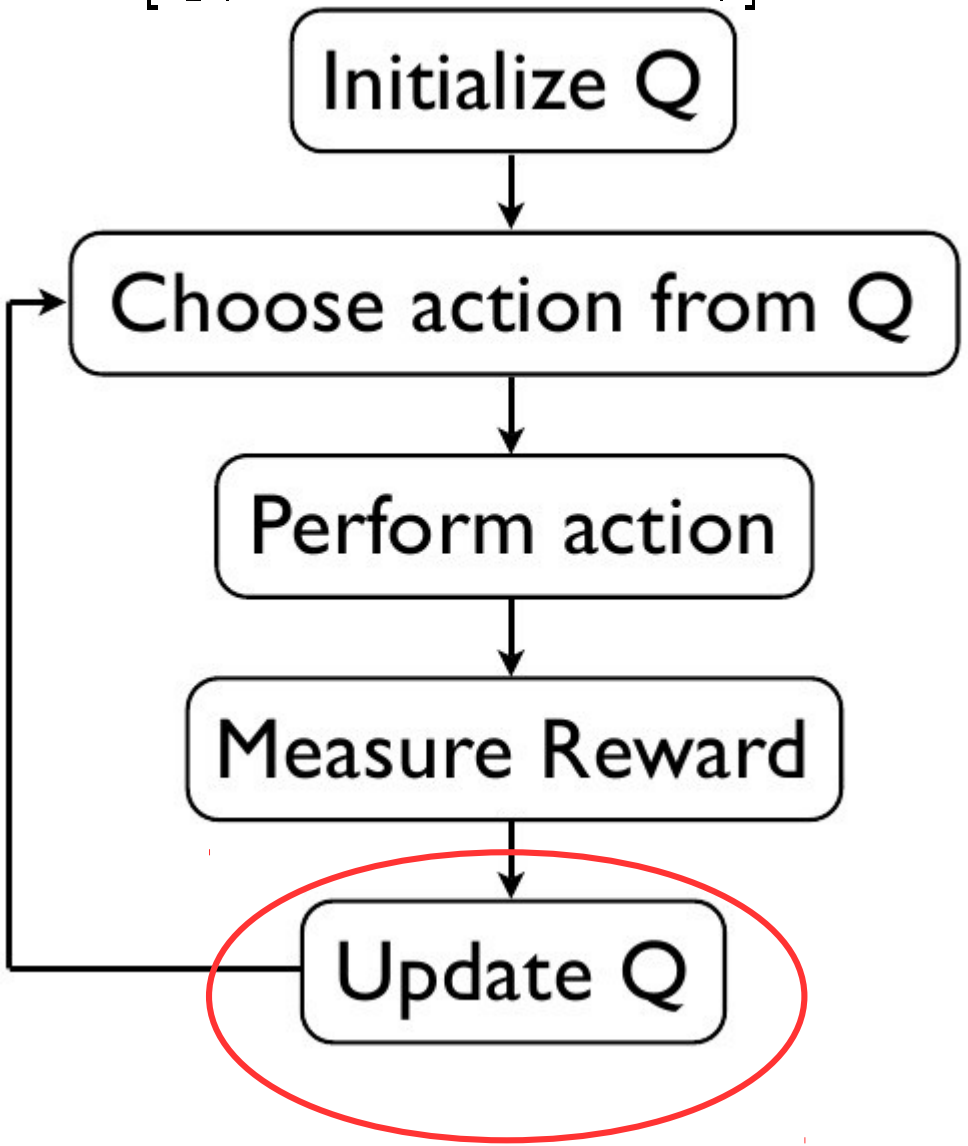
続けると。。。

$$
\mathbf{Q} = \begin{array}{c} state \backslash action \\ A \\ B \\ C \\ D \\ E \\ F \end{array}
\begin{array}{c}
\begin{array}{cccccc} A & B & C & D & E & F \end{array} \\
\begin{bmatrix}
- & - & - & - & 400 & - \\
- & - & - & 320 & - & 500 \\
- & - & - & 320 & - & - \\
- & 400 & 256 & - & 400 & - \\
320 & - & - & 320 & - & 500 \\
- & 400 & - & - & 400 & 500
\end{bmatrix}
\end{array}
$$

quiz1

$$\mathbf{Q}(state, action) = \mathbf{R}(state, action) + \gamma \cdot Max\left[\mathbf{Q}(next\ state, all\ actions)\right]$$

- quiz1.py
- この手順を書いてみる
- 難しいなら quiz11.py
  - 式を書いてみる

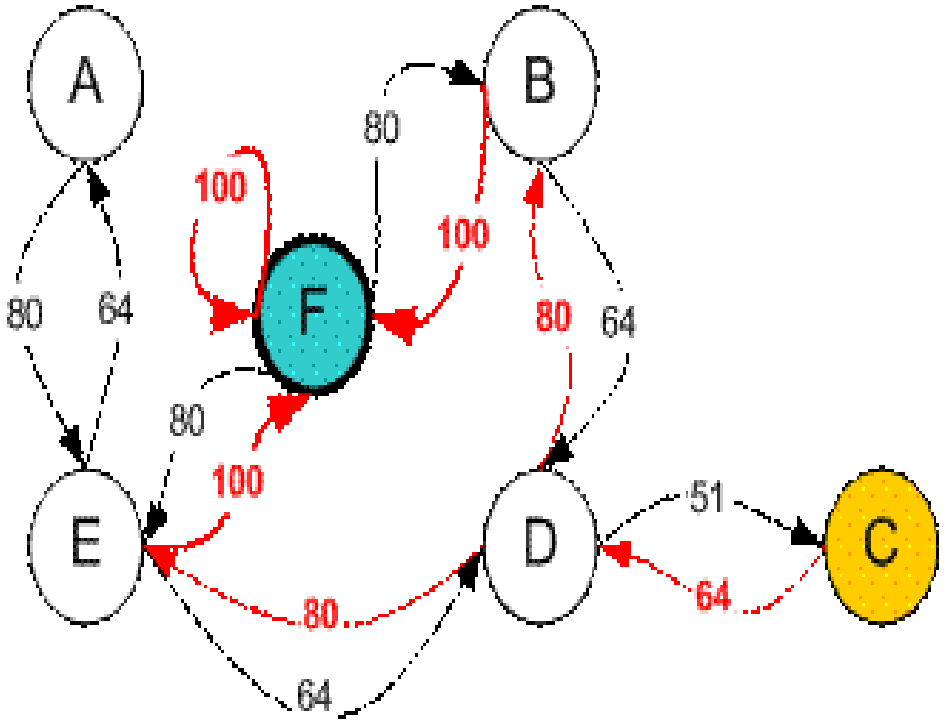


QTable を実装する

$$Q = \begin{array}{c} state \backslash action \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} - & - & - & - & 400 & - \\ - & - & - & 320 & - & 500 \\ - & - & - & 320 & - & - \\ - & 400 & 256 & - & 400 & - \\ 320 & - & - & 320 & - & 500 \\ - & 400 & - & - & 400 & 500 \end{bmatrix} \end{array}$$

$$\tilde{Q} = \begin{array}{c} state \backslash action \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} - & - & - & - & 80 & - \\ - & - & - & 64 & - & 100 \\ - & - & - & 64 & - & - \\ - & 80 & 51 & - & 80 & - \\ 64 & - & - & 64 & - & 100 \\ - & 80 & - & - & 80 & 100 \end{bmatrix} \end{array}$$

QTable を実装する



$$\tilde{Q} = \begin{array}{c} state \backslash action \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{bmatrix} - & - & - & - & 80 & - \\ - & - & - & 64 & - & 100 \\ - & - & - & 64 & - & - \\ - & 80 & 51 & - & 80 & - \\ 64 & - & - & 64 & - & 100 \\ - & 80 & - & - & 80 & 100 \end{bmatrix} \end{array}$$

quiz2

- quiz2.py
- QTable を使う実装プログラムを作ってみる。

quiz3

- quiz3.py
- 報酬 R を書いてみよう。