Science And Technology

特徴量関数近似 sarsa(λ)による 強化学習の汎用化

2016/05/11

mabonki0725

趣旨

Science And Technology

- Alpha-碁は強化学習モデルDQNを採用している
- DQNはゲームにも採用され もはや汎用モデルである

- DQNの前身であるsarsa(λ)の汎用性について述べる

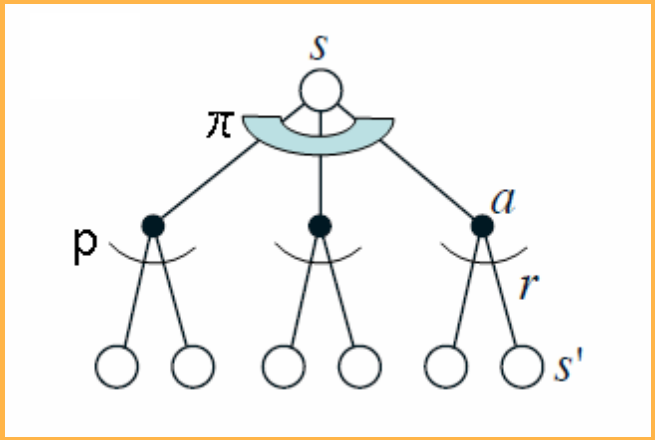
	B. Rider	Breakout	Enduro	Pong	Q*bert	Seaquest	S. Invaders
Random	354	1.2	0	−20.4	157	110	179
→ Sarsa [3]	996	5.2	129	−19	614	665	271
Contingency [4]	1743	6	159	−17	960	723	268
DQN	**4092**	**168**	**470**	**20**	**1952**	**1705**	**581**
Human	7456	31	368	−3	18900	28010	3690
HNeat Best [8]	3616	52	106	19	1800	920	**1720**
HNeat Pixel [8]	1332	4	91	−16	1325	800	1145
DQN Best	**5184**	**225**	**661**	**21**	**4500**	**1740**	1075

DQN論文(Playing Atari with Deep Reinforcement Learning)のベンチマーク表

強化学習

Science And Technology

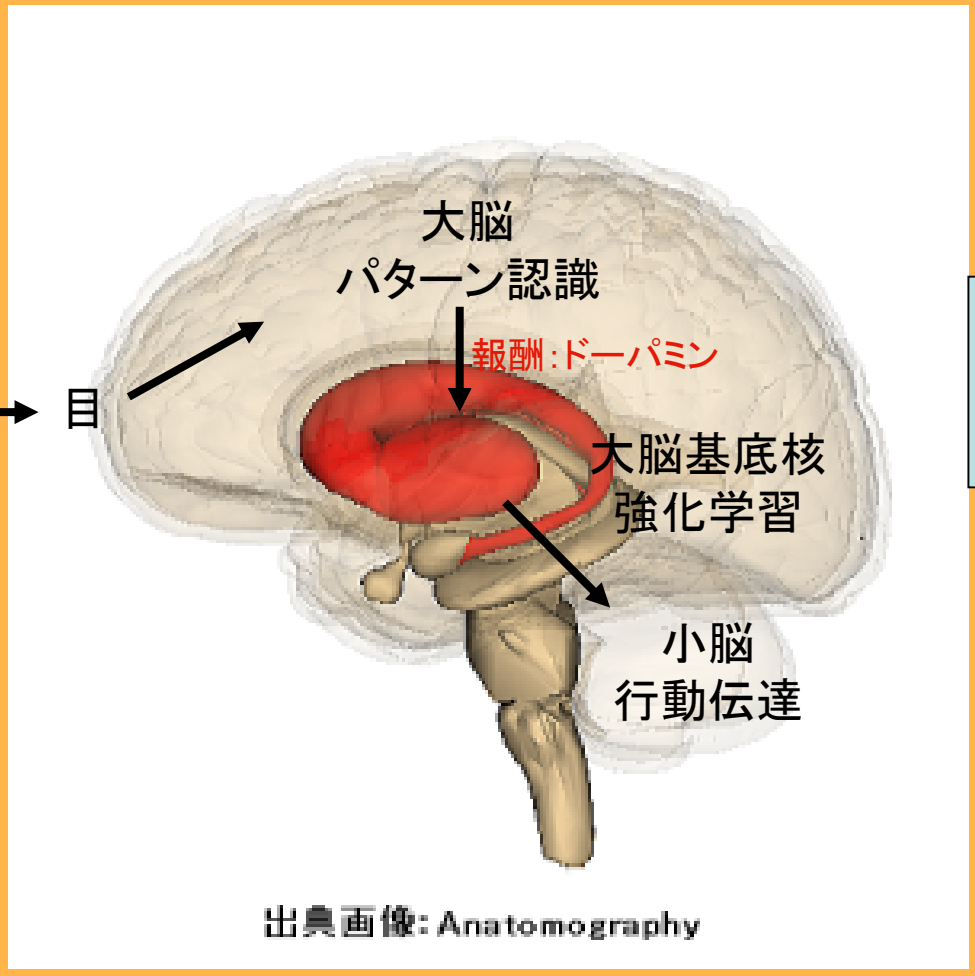
- 状況(s)で複数の行動(a)毎に報酬(r)があり、戦略(π)で行動を選択する
- 行動(a)の選択は現在の状況(s)のみで決定される(マルコフ決定過程)
- 状況(s)で将来の報酬(r)の合計が最大と**予測**する行動(a)を選択する
- 戦略(π)によって異なる行動(a)採る(強気、弱気、経験的、冒険的等)



強化学習は次の行動を決めるため、
将来の報酬の合計である
価値関数V(s) OR 行動価値関数Q(s,a)の算出が目的

脳と強化学習の類似

Science And Technology



大脳基底核の脳波と強化学習の価値関数のプロット図が同形を示す

銅谷賢治

価値関数V 行動関数Qの算出方法

Science And Technology

- 将来の状況と行動の分岐を繰返し展開して末端からBackUpして関数を算出する（rollout）
- 将来価値にγ:減少率があるので無限に展開しなくて良い(n期先まで展開)。
  - 動的計画法 （遷移を定常になるまで繰返し）
  - ﾓﾝﾃｶﾙﾛ法 （ﾗﾝﾀﾞﾑに経路を辿り出現確率で計算）
  - TD(λ)法 （V関数のパラメータをSDGで計算）
  - Sarsa(λ)法 （Q関数のパラメータをSDGで計算）
  - ニューロ法 （Q関数をニューロで汎用化）
  - DQN （DeepLearningで特徴量を抽出して計算）

Science Technology

特徴量による価値関数近似 学習を繰返し最適なAとwを算出

価値関数を特徴量と重みで定義した場合、その重みは以下で更新される

$$\bar{Q}(s,a,w) = w^T x(s,a) = \sum_{i=1}^{n} w_i x_i(s,a)$$

Xi：特徴量（s:状態、a:行動)

$$w_{t+1} = w_t - \frac{1}{2}\alpha \nabla_w \left[Q_\pi(S_t,A_t) - \bar{Q}(S_t,A_t,w_t)\right]^2$$

SGD

$$= w_t + \alpha\left[Q_\pi(S_t,A_t) - \bar{Q}(S_t,A_t,w_t)\right]\nabla_w \bar{Q}(S_t,A_t,w_t)$$

二乗誤差の微分

**学習を繰返す毎に
Qを最大化する行動Aを学習し
重みwを最適化する**

Sarsa(λ)では重みは以下で更新される

$$w_{t+1} = w_t + \alpha\delta_t z_t$$

ここで

$$\delta_t = R_{t+1} + \gamma\bar{Q}(S_{t+1},A_{t+1},w_t) - \bar{Q}(S_t,A_t,w_t)$$

$$z_t = \gamma\lambda z_{t-1} + \nabla_w \bar{Q}(S_t,A_t,w_t)$$

である

過去の微分値の蓄積(λ：重み)

特徴量による価値関数近似 sarsa(λ) アルゴリズム

Science And Technology

Initialize $\mathbf{w}$ as appropriate for the problem, e.g., $\mathbf{w} = \mathbf{0}$
Repeat (for each episode):
  $\mathbf{z} = 0$
  $S \leftarrow$ initial state of episode
  Repeat (for each step of episode):
    $A \leftarrow$ action given by π for S
    Take action A, observe reward, R, and next state, S'
    $\delta \leftarrow R + \gamma \hat{v}(S', \mathbf{w}) - \hat{v}(S, \mathbf{w})$
    $\mathbf{z} \leftarrow \gamma\lambda\mathbf{z} + \nabla_{\mathbf{w}} \hat{v}(S, \mathbf{w})$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\mathbf{z}$
    $S \leftarrow S'$
  until S' is terminal

強化学習の汎用化

Science And Technology

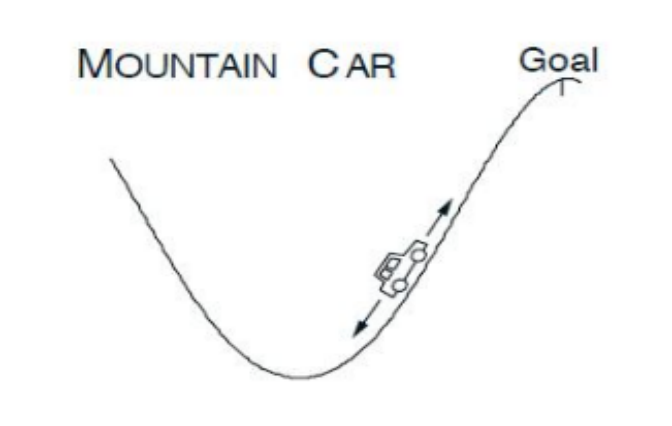
- 問題毎に特徴量のみ設定する
- 局面毎に報酬を設定しなくてよい
  目標や障害との位置によって適切な報酬を決めるのは困難

- 下記のみ設定すれば汎用的に解ける
  ①行動の設定
  ②特徴量の設定
  ③行動後の特徴量の変化
  ④初期条件
  ⑤終了条件

(例1)馬力不足の車の登り学習

Science And Technology

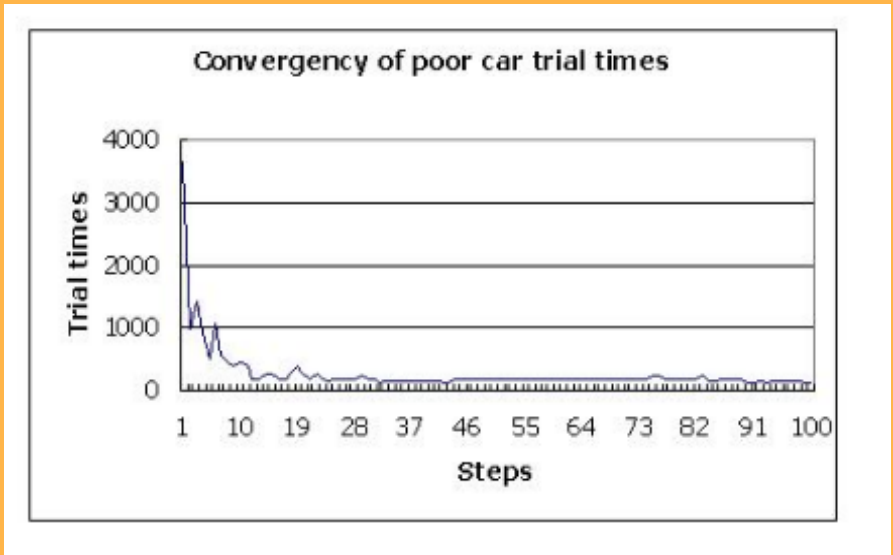
馬力不足の車があり後退、前進を繰返して
下降時の加速度を利用して坂登りを学習します

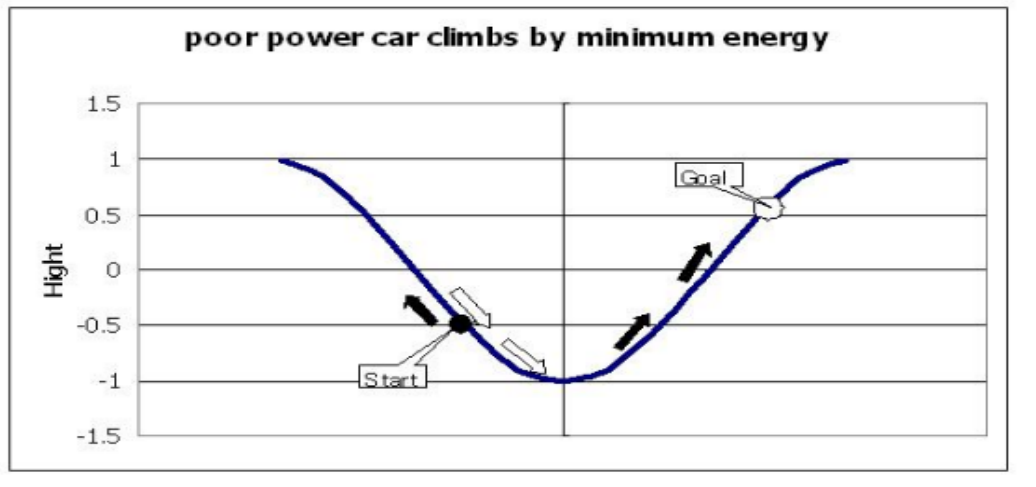


		汎用プラットホームに設定する値
①	行動	前進　後進　自由降下
②	特徴量	位置P　速度V
③	行動後の特徴量	前進：P=P+V　後進：P=P−V　V=C1−sin(P*C2)
④	初期条件	出発点
⑤	終了条件	終点に達する

(例1)馬力不足の車の登り学習

Science And Technology

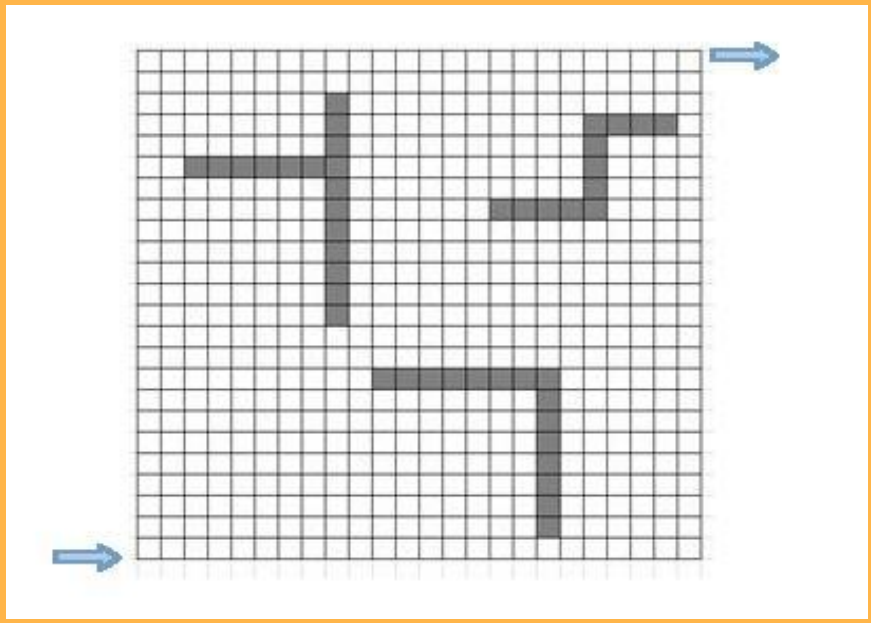




最初は4000回で達するが最後は3回の操作で登る

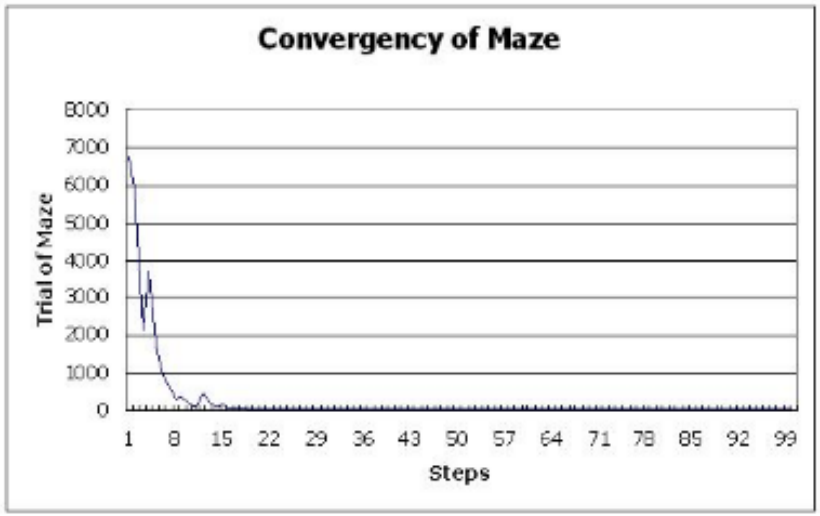
(例2)壁の障害を避ける学習

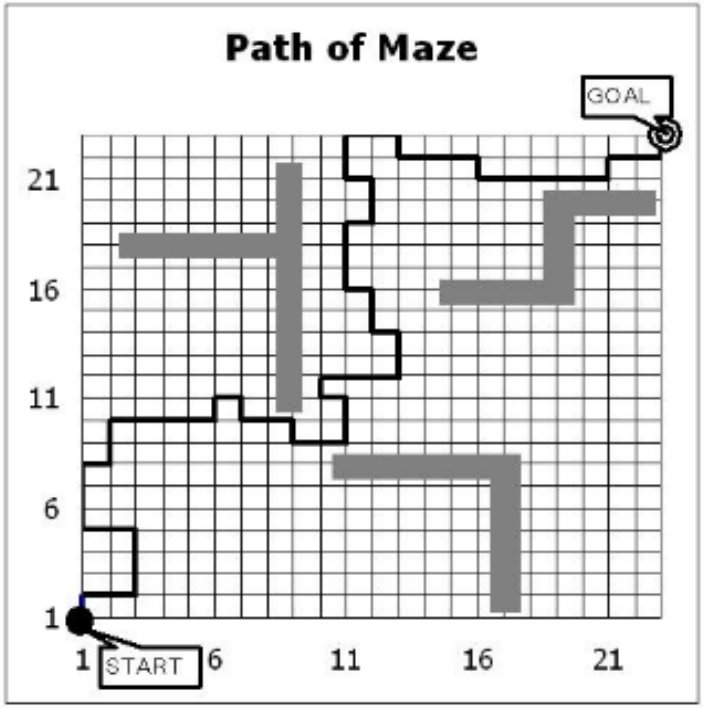
Science And Technology



汎用プラットホームに設定する値		
①	行動	㊧ ㊨ ㊤ ㊦
②	特徴量	横座標X　縦座標Y
③	行動後の特徴量	㊧x +=1　㊨x−=1 ㊤y+=1 ㊦y−=1　　但し壁は通れない
④	初期条件	左下隅
⑤	終了条件	右上隅に到着

(例2)壁の障害を避ける学習





最初は7000回かかるが最後は60回で出口に達する

まとめ

Science And Technology

- 強化学習sarsa(λ)の特徴量近似モデルは全く別の問題を特徴量の指定のみによって汎用的に解けることを示した
- 逆に適切な特徴量の指定が大事
  - DQN=DeepLearning(特徴量抽出)＋強化学習
- 報酬を設定する必要がないことを示した
  - 局面毎に適切な報酬を設定した方が学習は早くなるが一般に決定には困難(ゲーム以外)
    →価値関数から報酬を逆算する逆強化学習がある

参考文献

Science And Technology

- Mastering the Game of Go with Deep Neural Network and Tree Seach *DeepMind*
- Playing Atari with Deep Reinforcement Learning *DeepMind*
- Reinforcement Learning *Sutton*
- 心の分子機構への計算理論的アプローチ *銅谷 賢治*
- Probablistic Robtics *Thurn*
- Maximum Entropy Deep Inverse Reinforcement Learning *NIPS2014*
- Inverse Reinforcement Learning with Locally Consistent Reward Functions *(NIPS2015)*