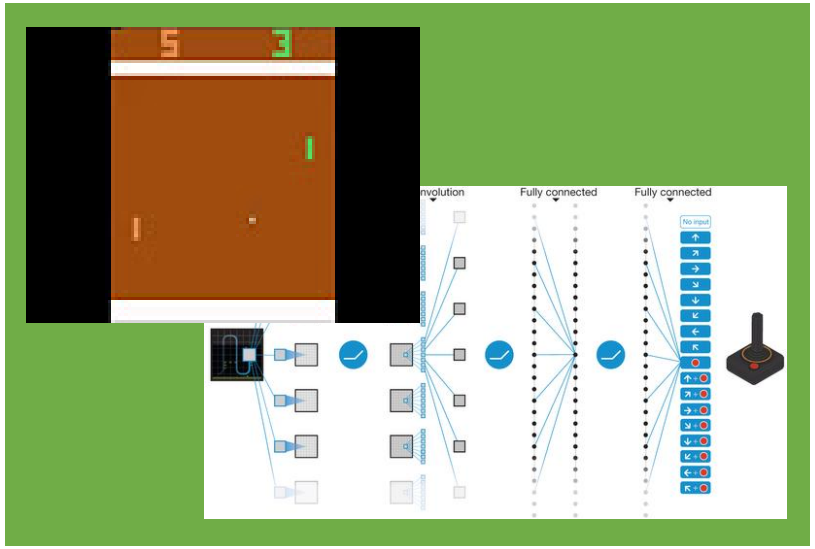
強化学習

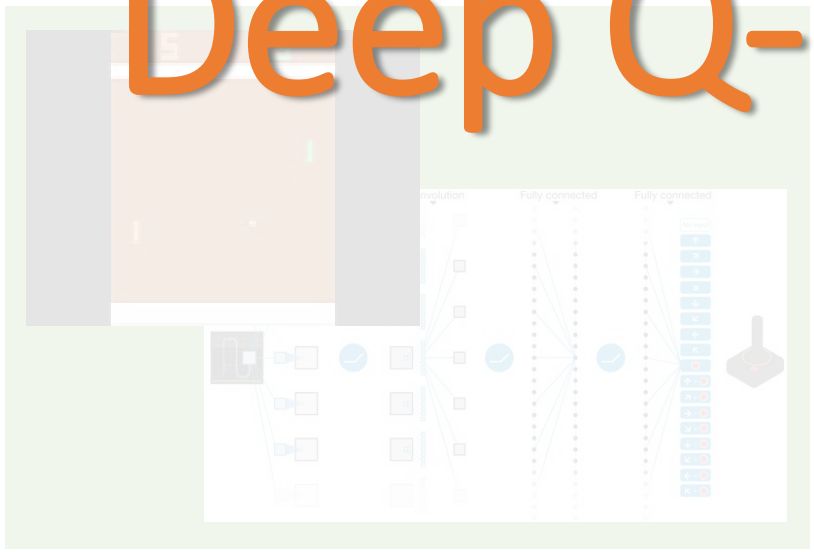
本日お話しすること


AlphaGo
Lee Sedol


5
3
Convolution
Fully connected
Fully connected

本日お話しすること

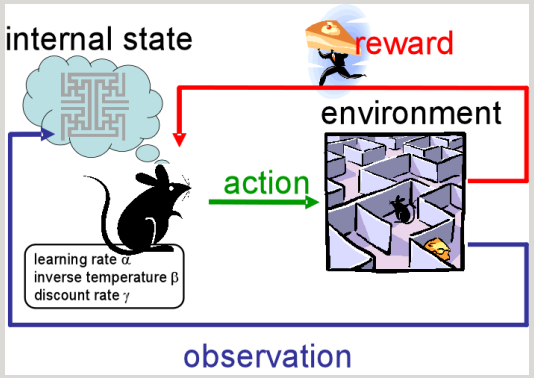
Deep Q-Network



本日お話しすること

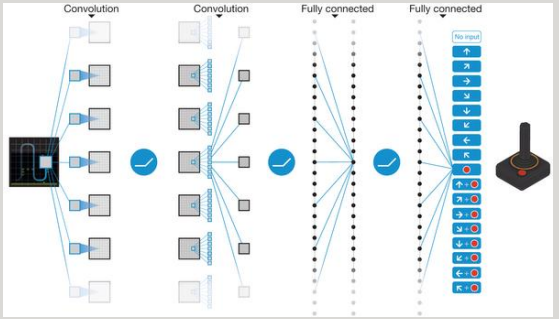
Chapter

前半



強化学習

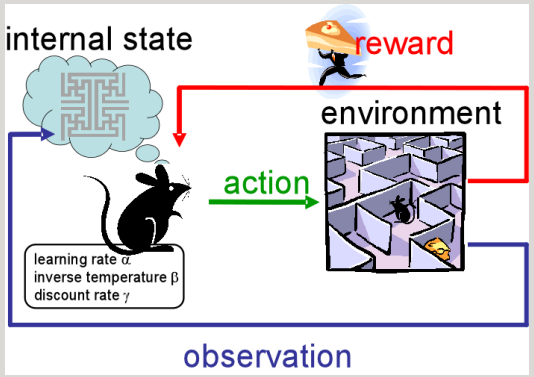
後半



Deep Q-Network

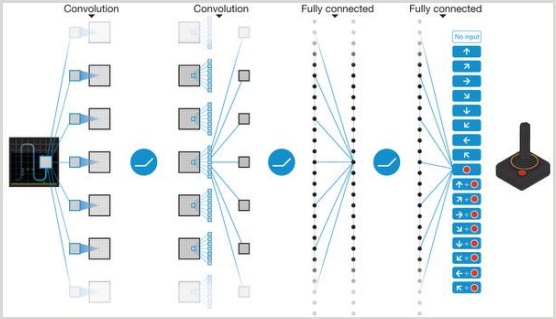
Chapter

前半



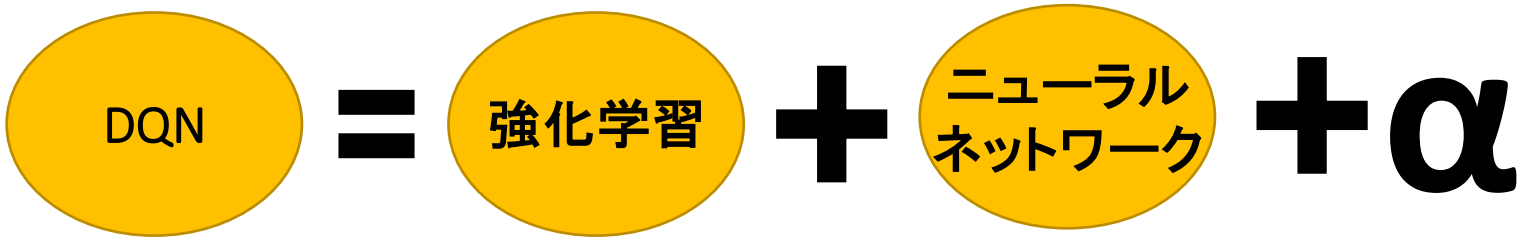
強化学習

後半

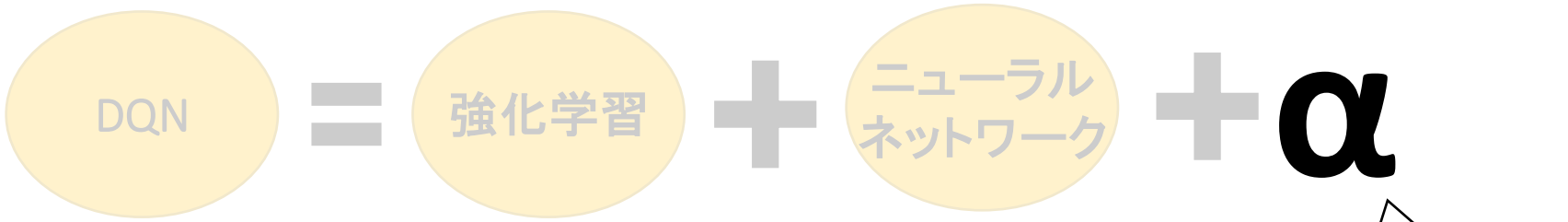


Deep Q-Network

DQNで使われている技術



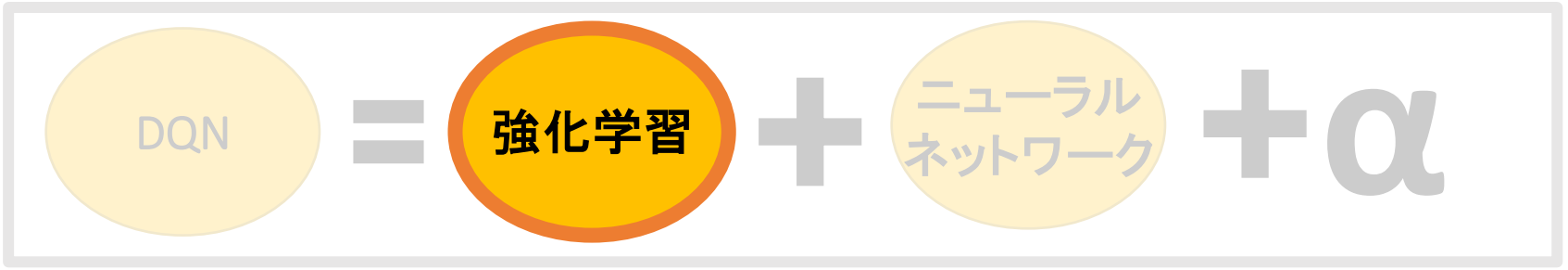
DQNで使われている技術



- Experience Replay
- Target Q-Network
- Clipping

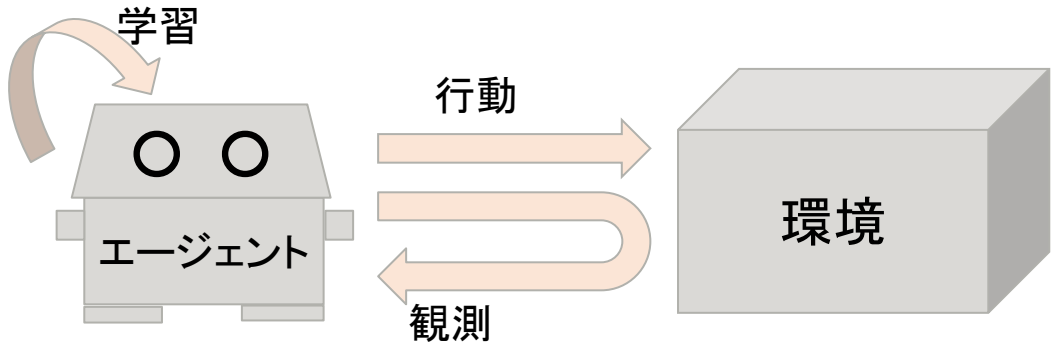
後半
“Deep Q-Network”
で説明

DQNで使われている技術



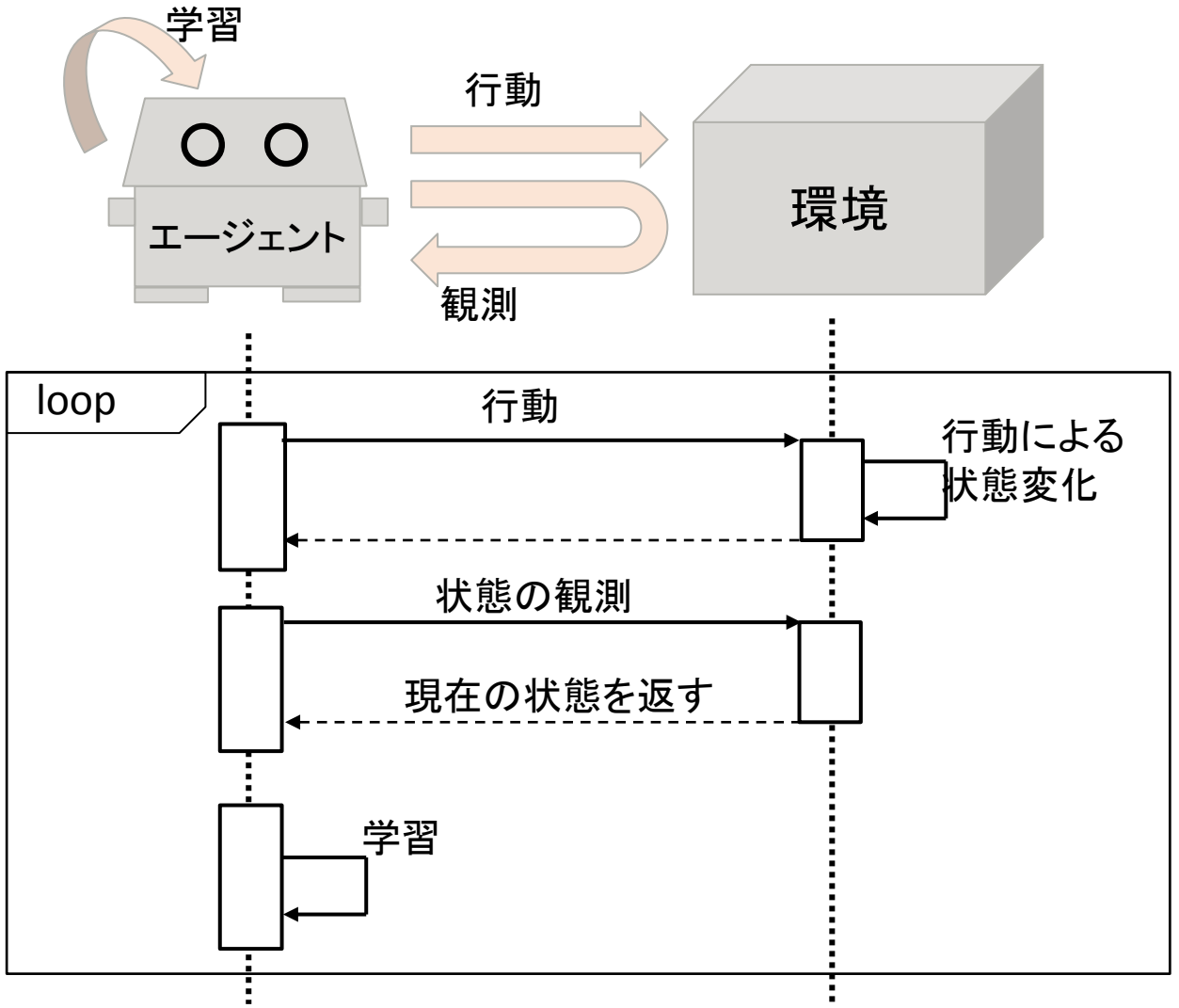
強化学習とは？

- エージェントが環境との相互作用を通して最適な方策を得る手法



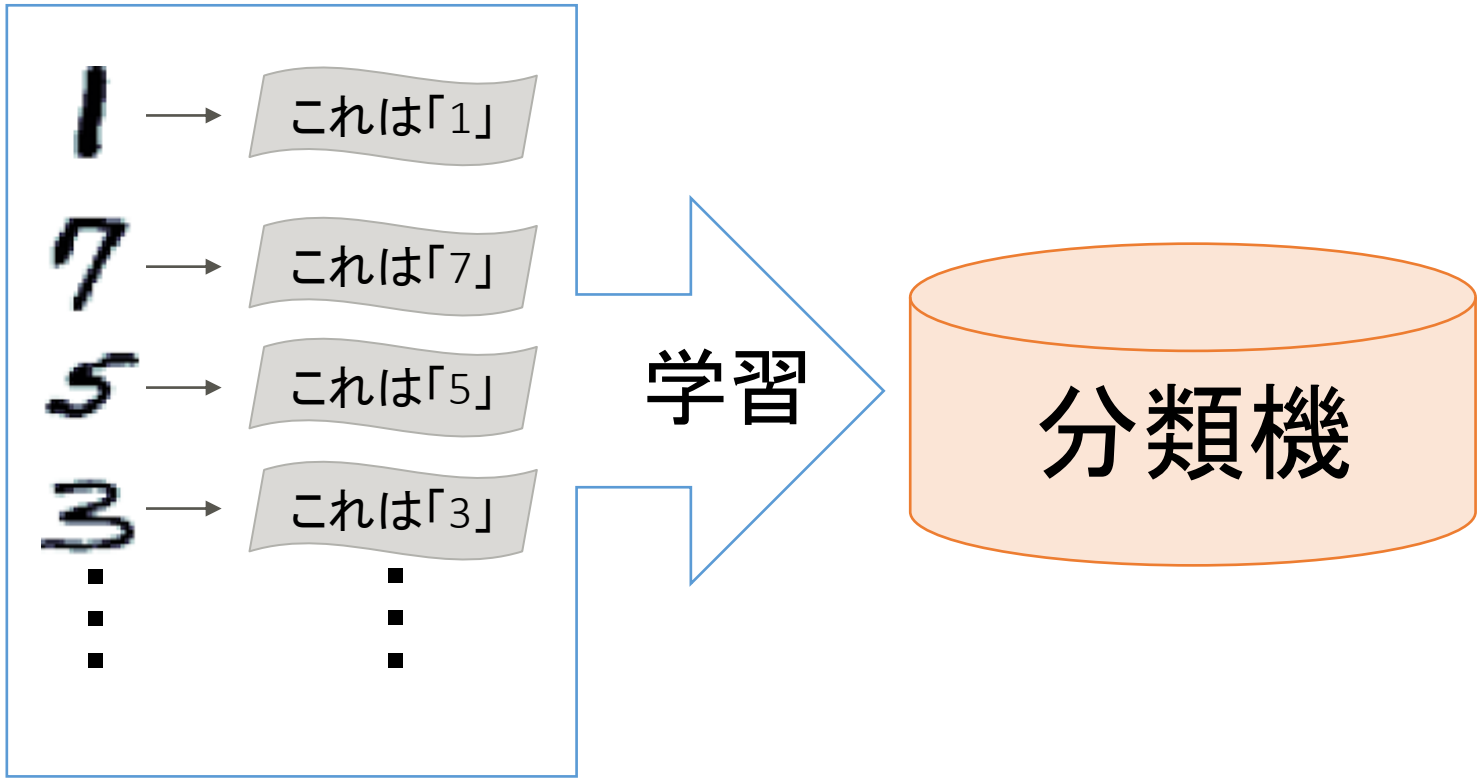
強化学習とは？

- エージェントが環境との相互作用を通して最適な方策を得る手法



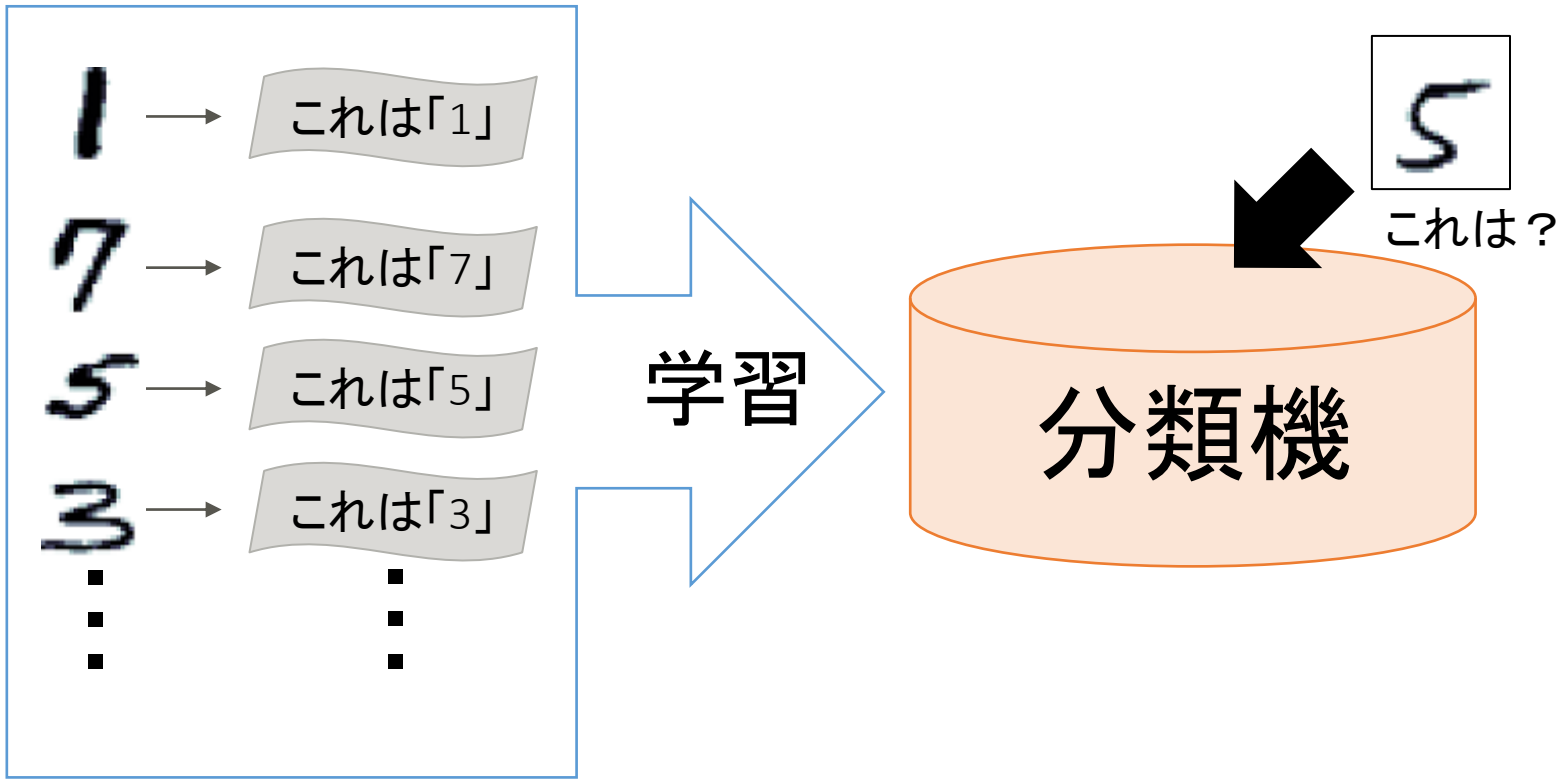
教師あり学習では…

- 与えられた正解に基づいて学習



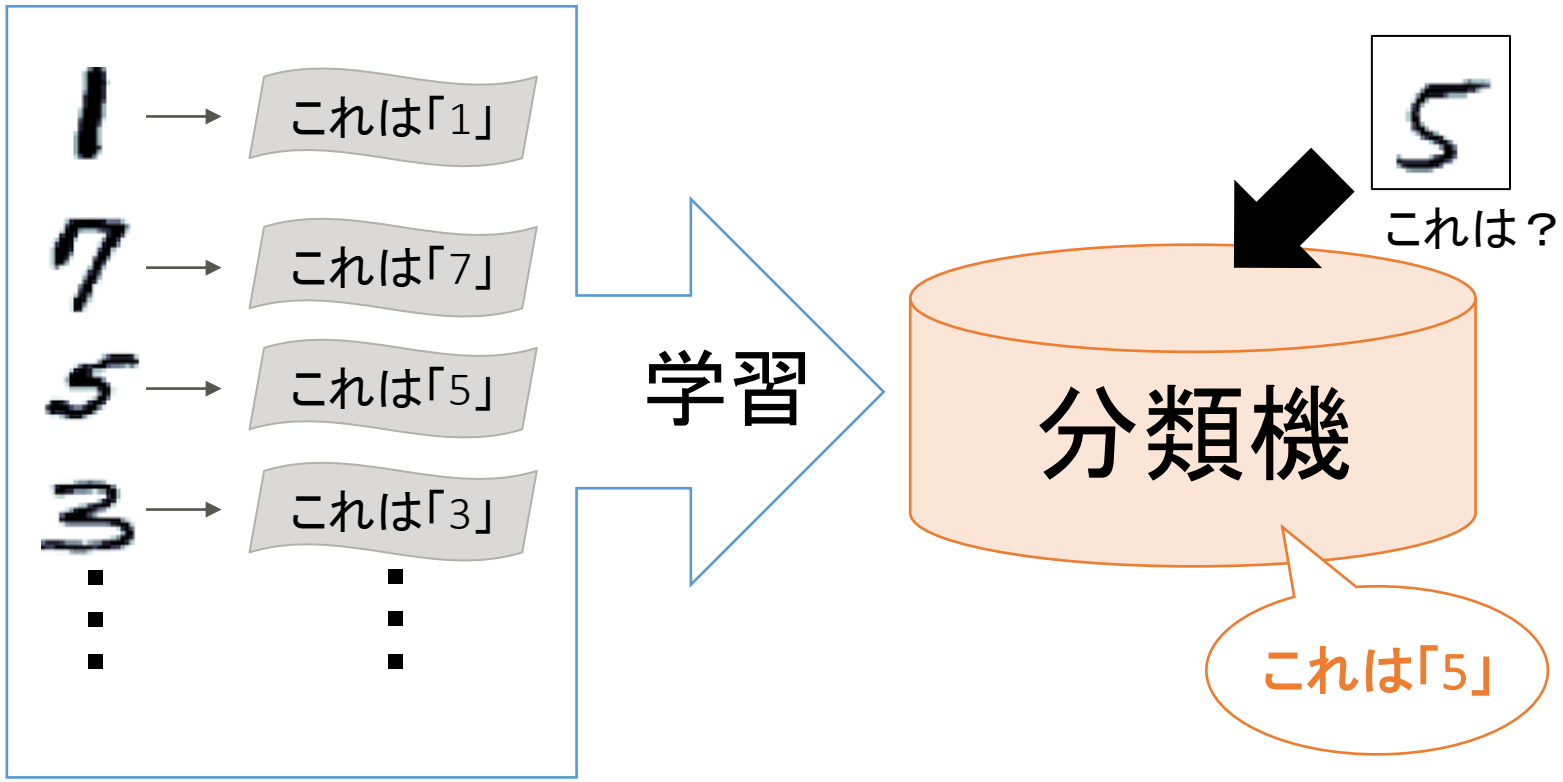
教師あり学習では…

- 与えられた正解に基づいて学習



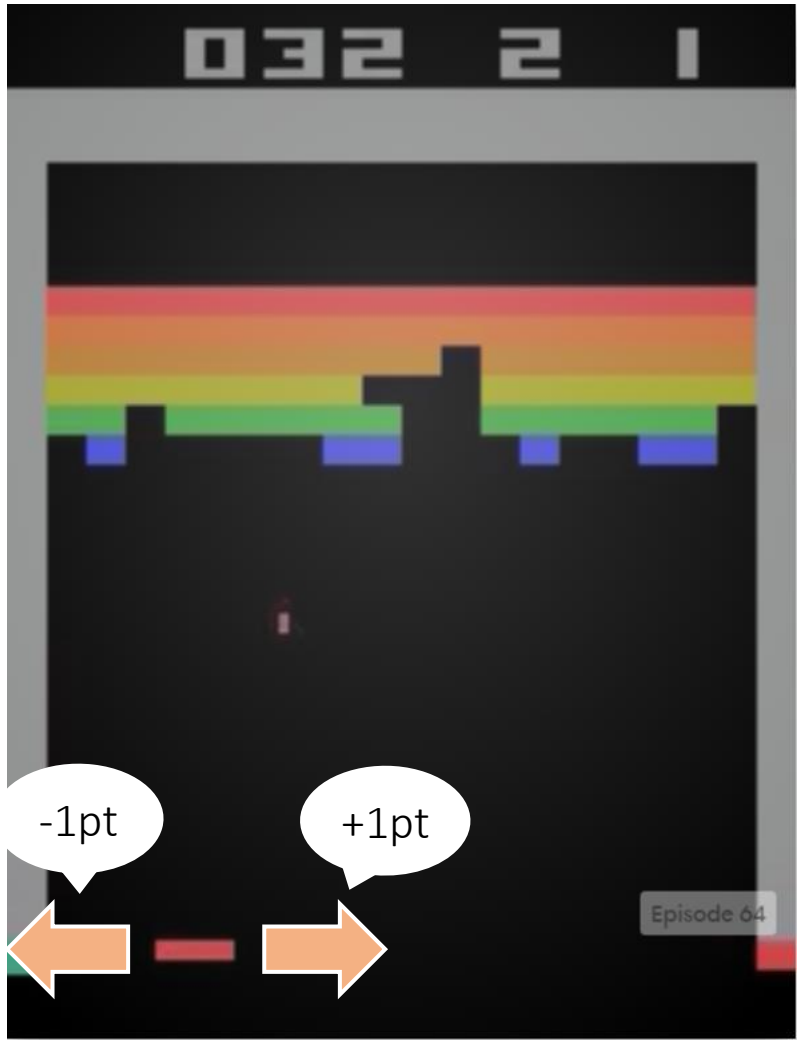
教師あり学習では…

- 与えられた正解に基づいて学習



強化学習では

- 明確な正解が与えられない



与えられるのは

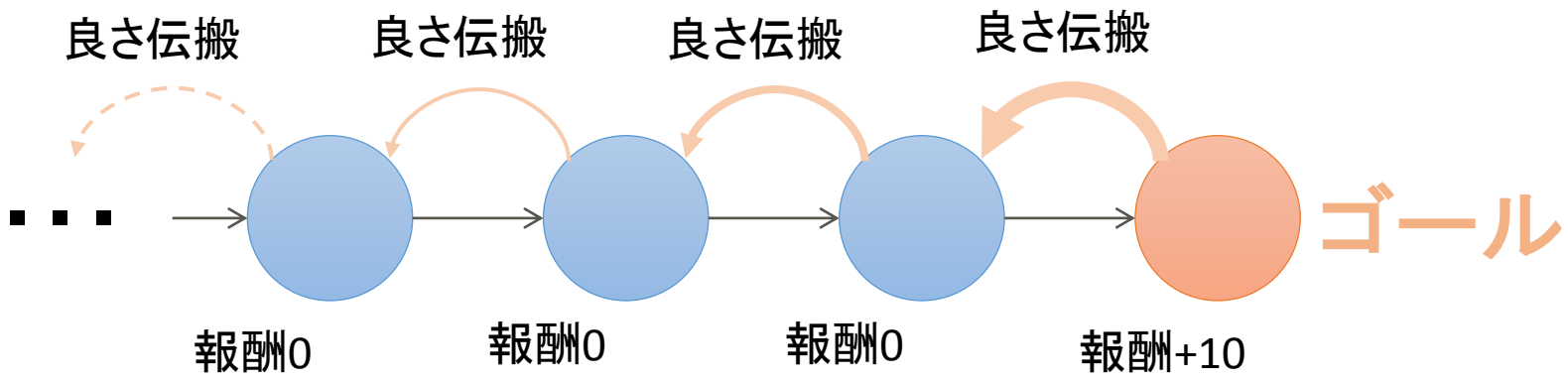
- 行動の選択肢
- 行動の良さの***見込み***

▽

正解ではない

行動の良さの見込みとは？

- (例) 迷路
  - 何手か動かして正解がわかる



Q学習では良さはQ値と呼ばれ、下式で更新される

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma max Q(\acute{s}) - Q(s,a)\right)$$

学習率
(主に0.1～1)

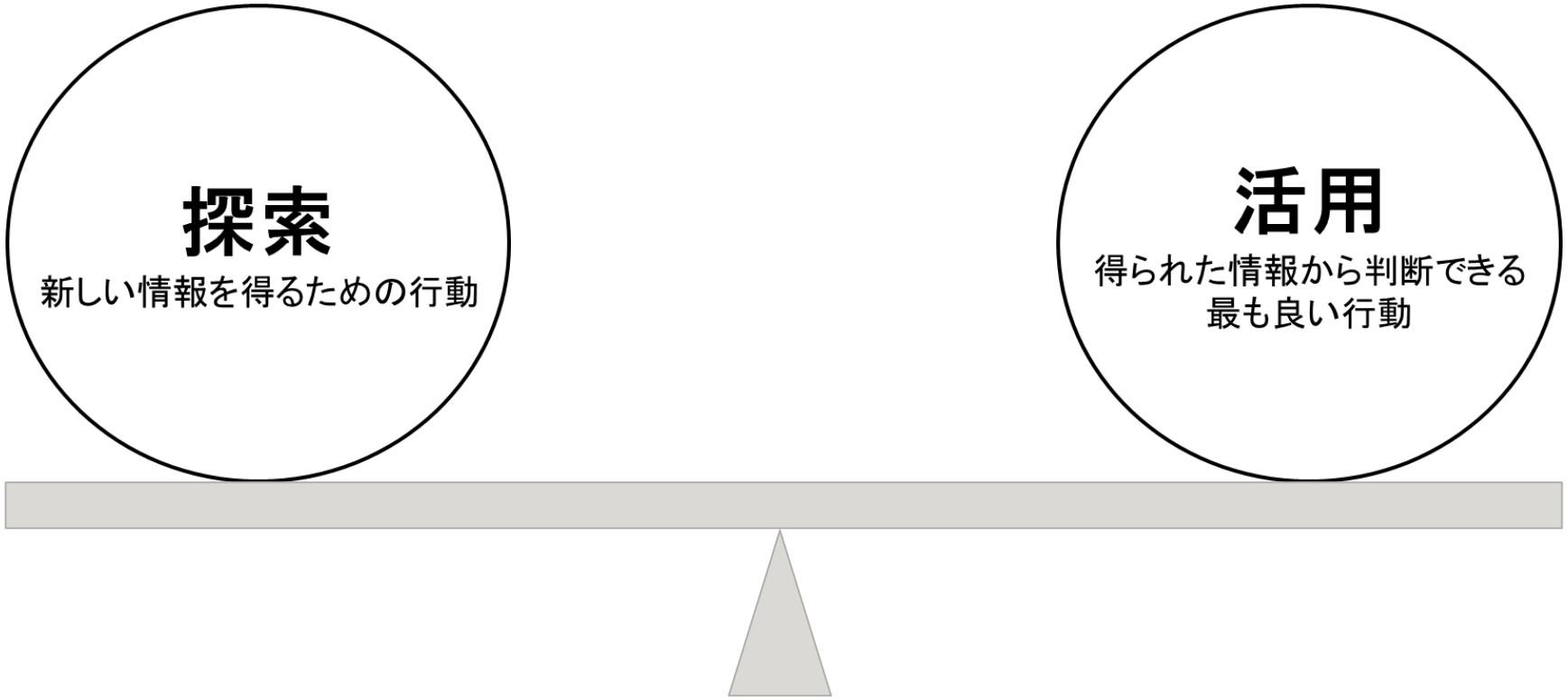
割引率
(主に0.9～0.99)

Q値の更新を眺めてみる

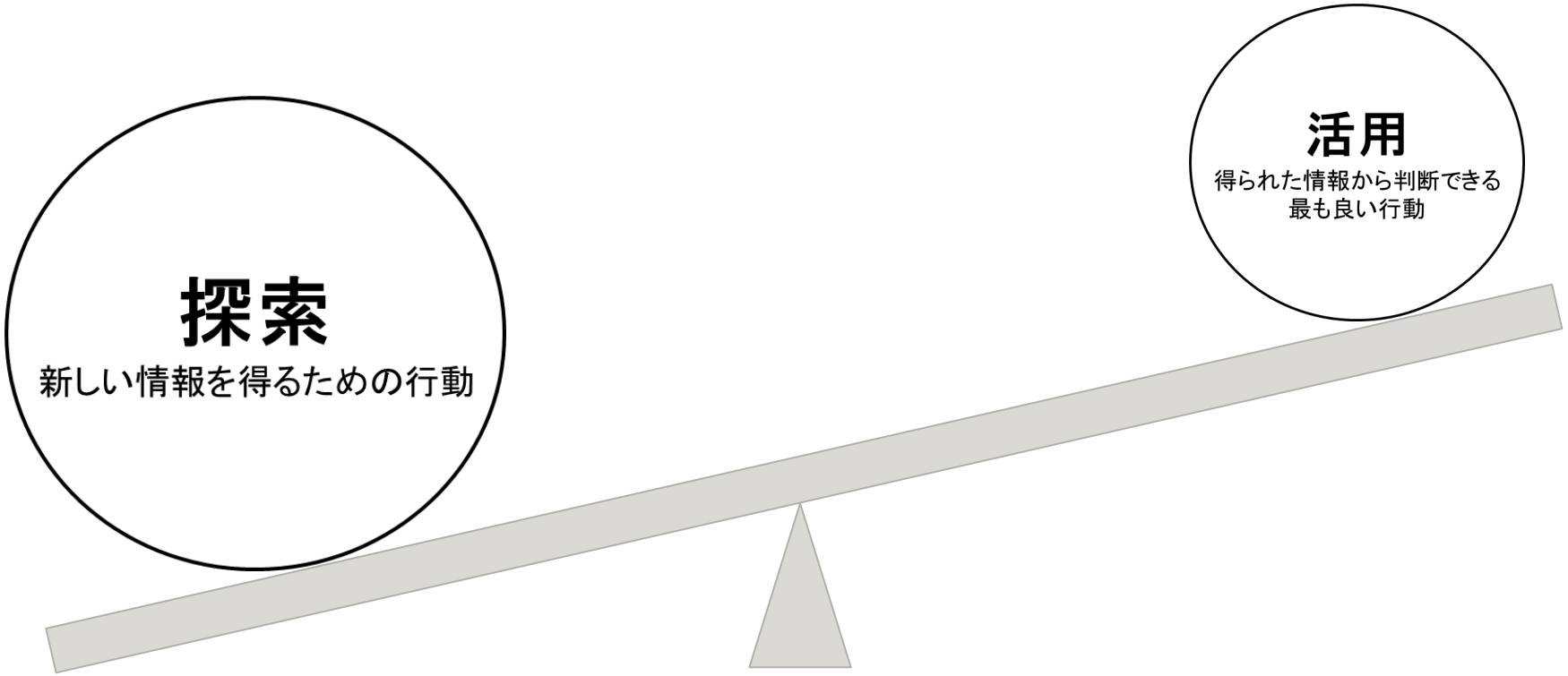
Qiita
http://qiita.com/hogefugabar/items/74bed2851a84e978b61c

Youtube
https://www.youtube.com/watch?v=-JXxYZ5HB8U

行動の種類

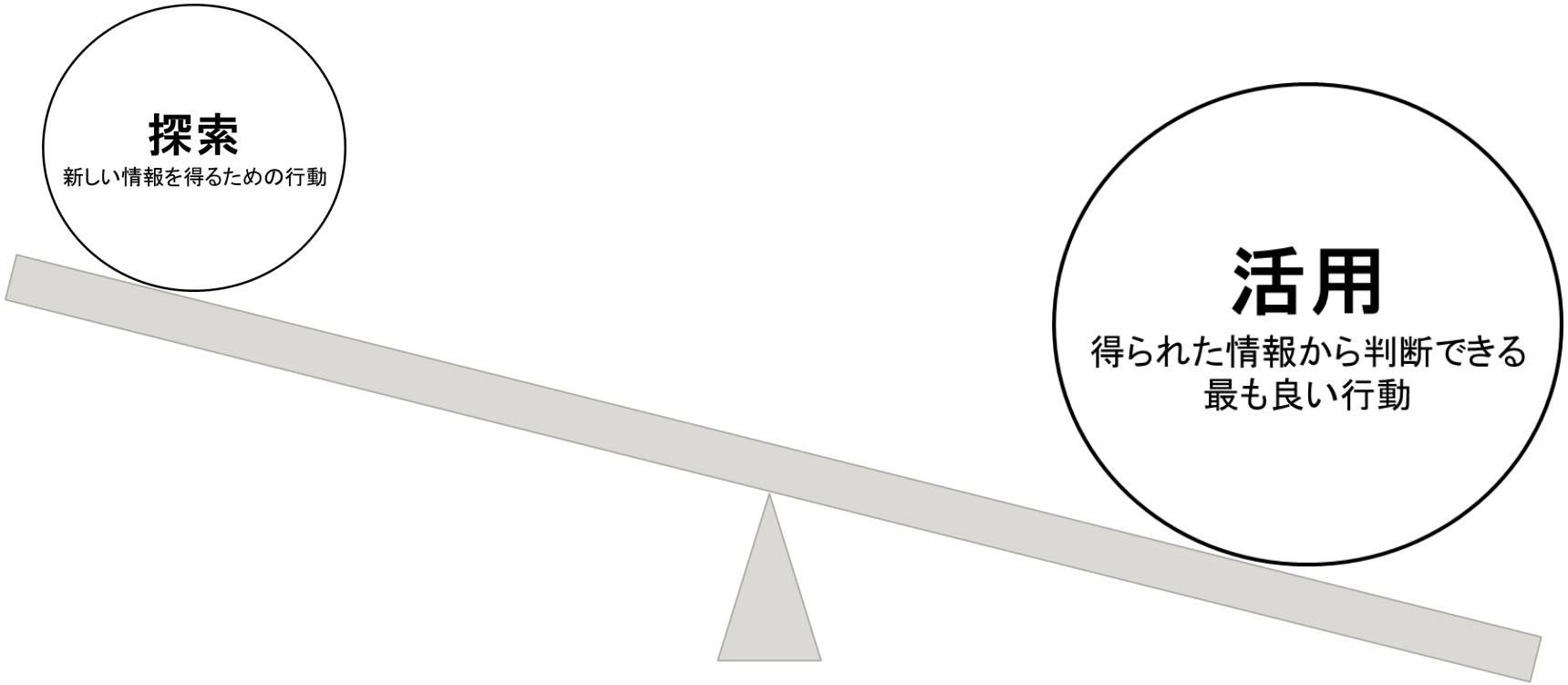


探索をしすぎると…



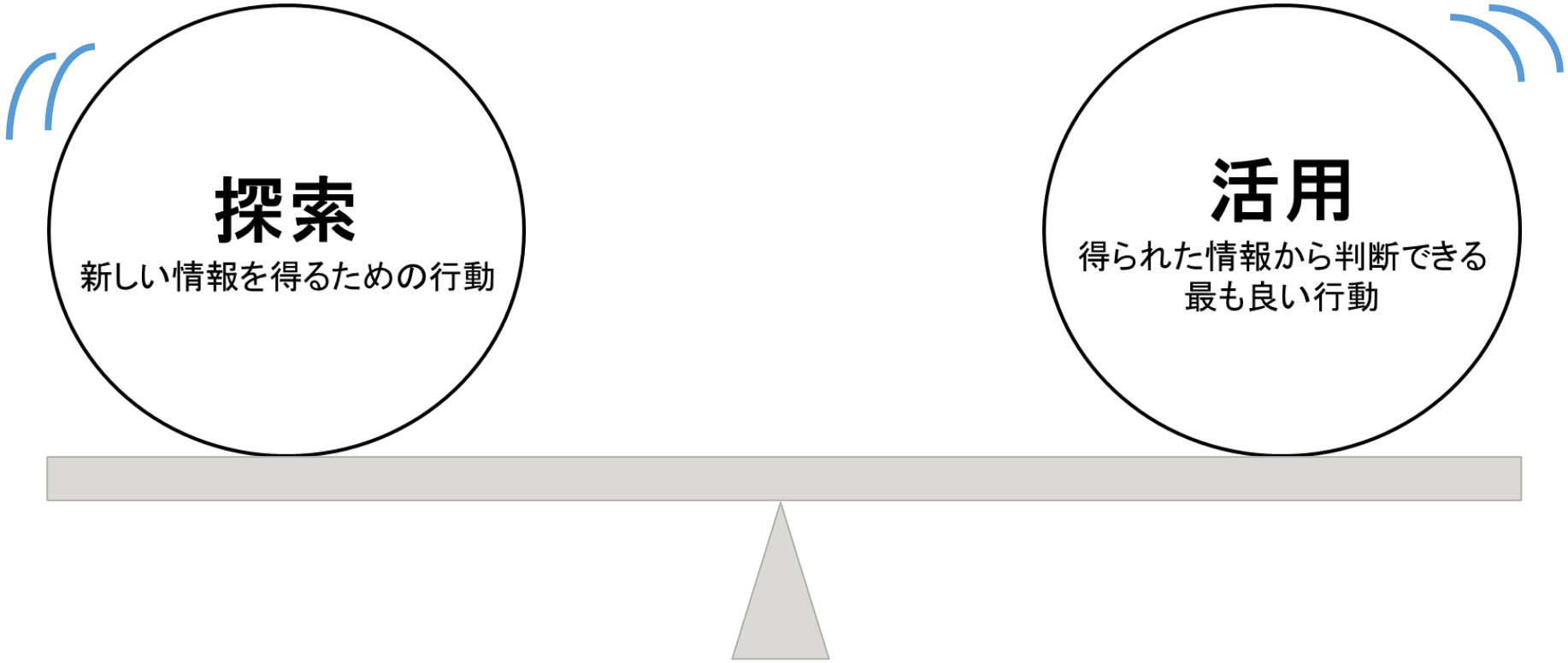
良い行動がわかっているのにその行動を選ばない

活用をしすぎると…



より良い行動に気付かない

バンディットアルゴリズム



探索と活用のトレードオフを解決

ε-greedy法

- 確率εで活用、確率(1-ε)で探索をする

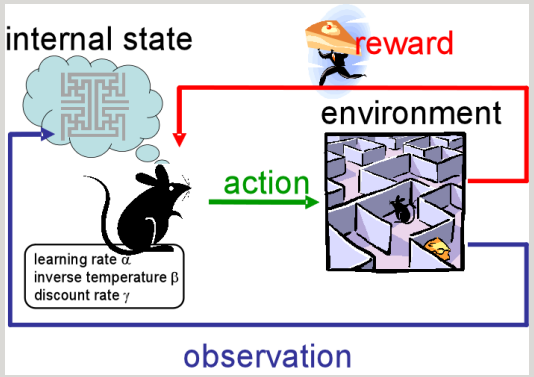
```
epsilon = 0.1
while True:
    x = random(0, 1)
    if epsilon > x:
        randomAction()
    else:
        bestAction()
```

```
epsilon = 1
while True:
    x = random(0, 1)
    if epsilon > x:
        randomAction()
    else:
        bestAction()
    epsilon -= 0.0001
```

DQNでよく使われるパターン

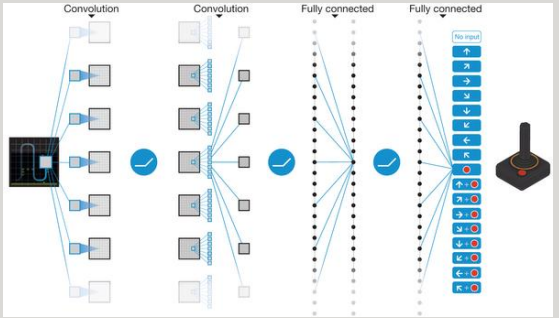
Chapter

前半



強化学習

後半



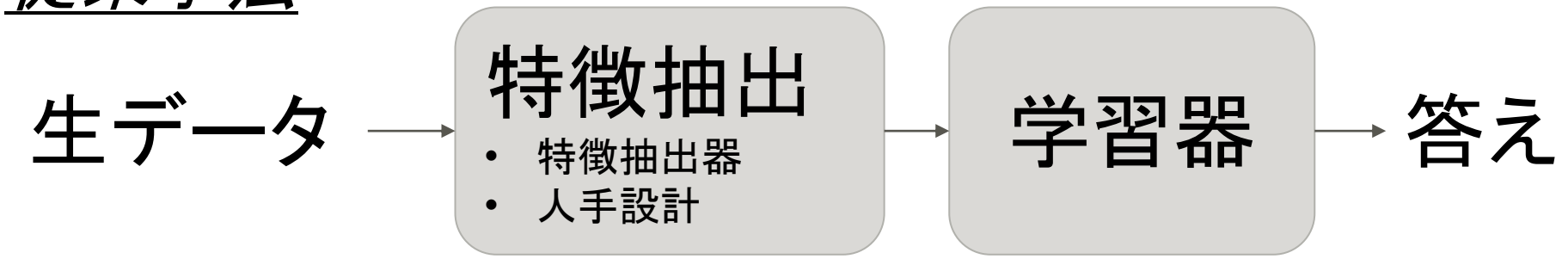
Deep Q-Network

DQNで使われている技術

Deep Learningで何が変わったのか

- ざっくりいうと、*”特徴量を細かく設定する必要がなくなった”*

従来手法



Deep Learning

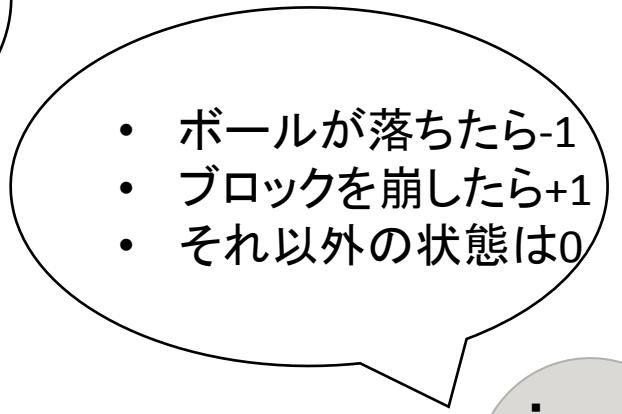
ニューラルネットワーク

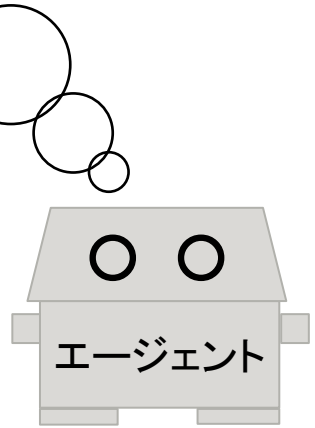
- A Neural Network Playground

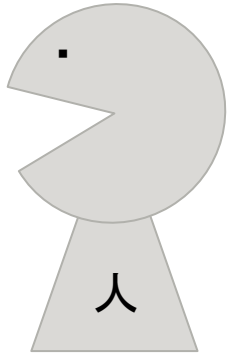
A Neural Network Playground
http://playground.tensorflow.org/

なにがすごい？

- 世界の仕組みを細かく教えなくてよい



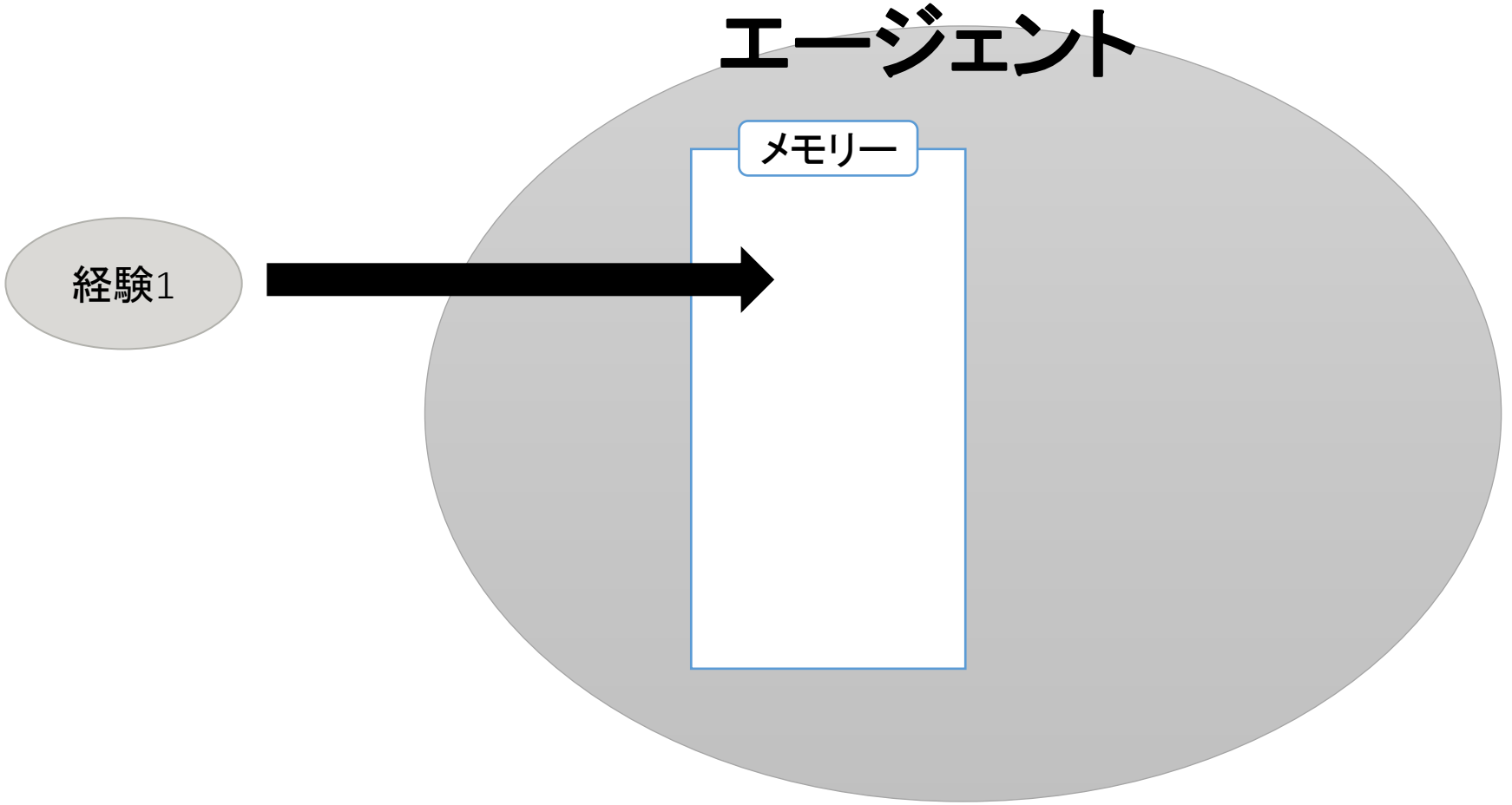




DQNで使われている技術

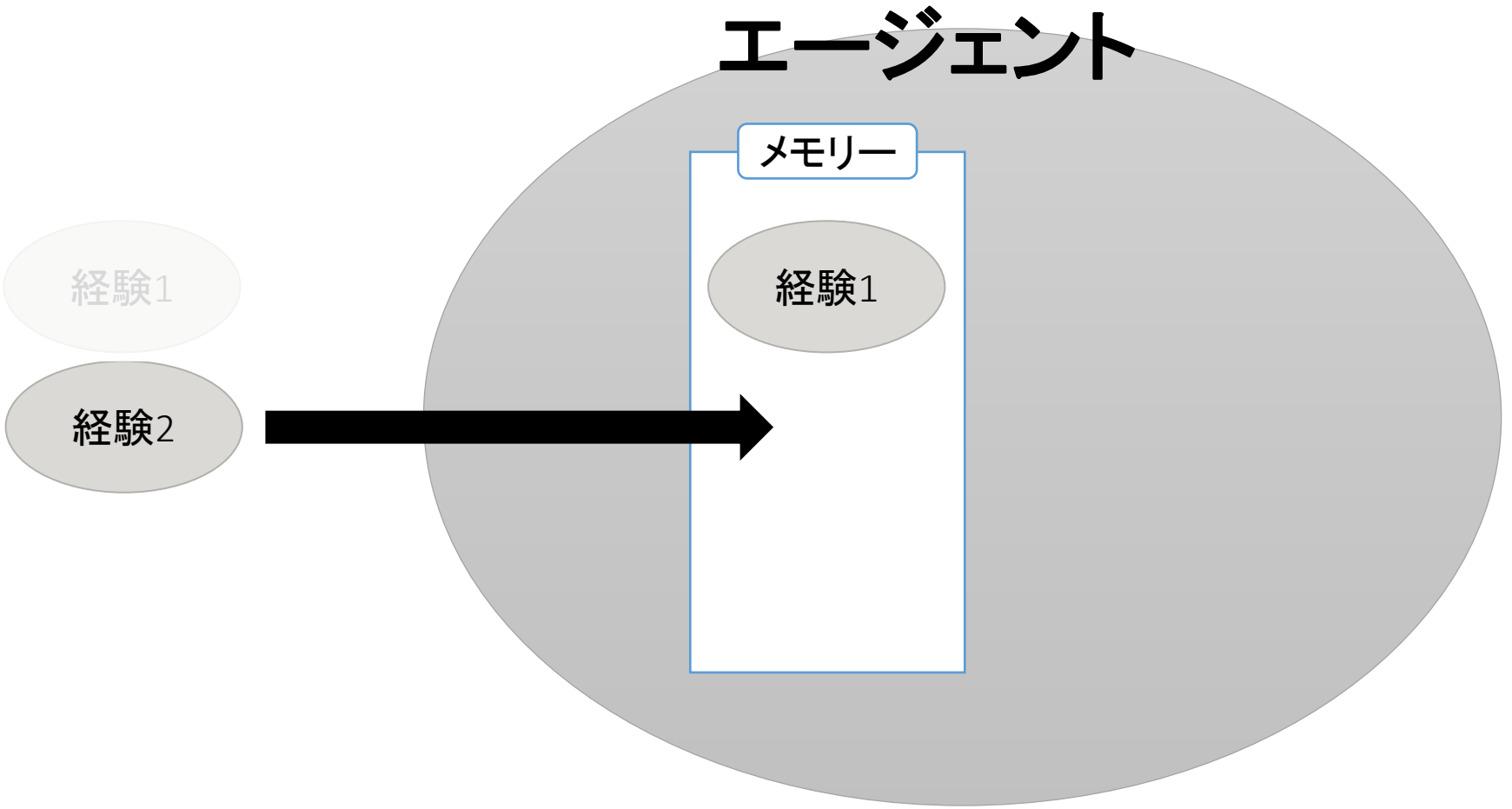
Experience Replayのイメージ

- 経験(状態、行動、報酬、遷移)をメモリーに蓄積し、メモリーからランダムサンプリングして学習する



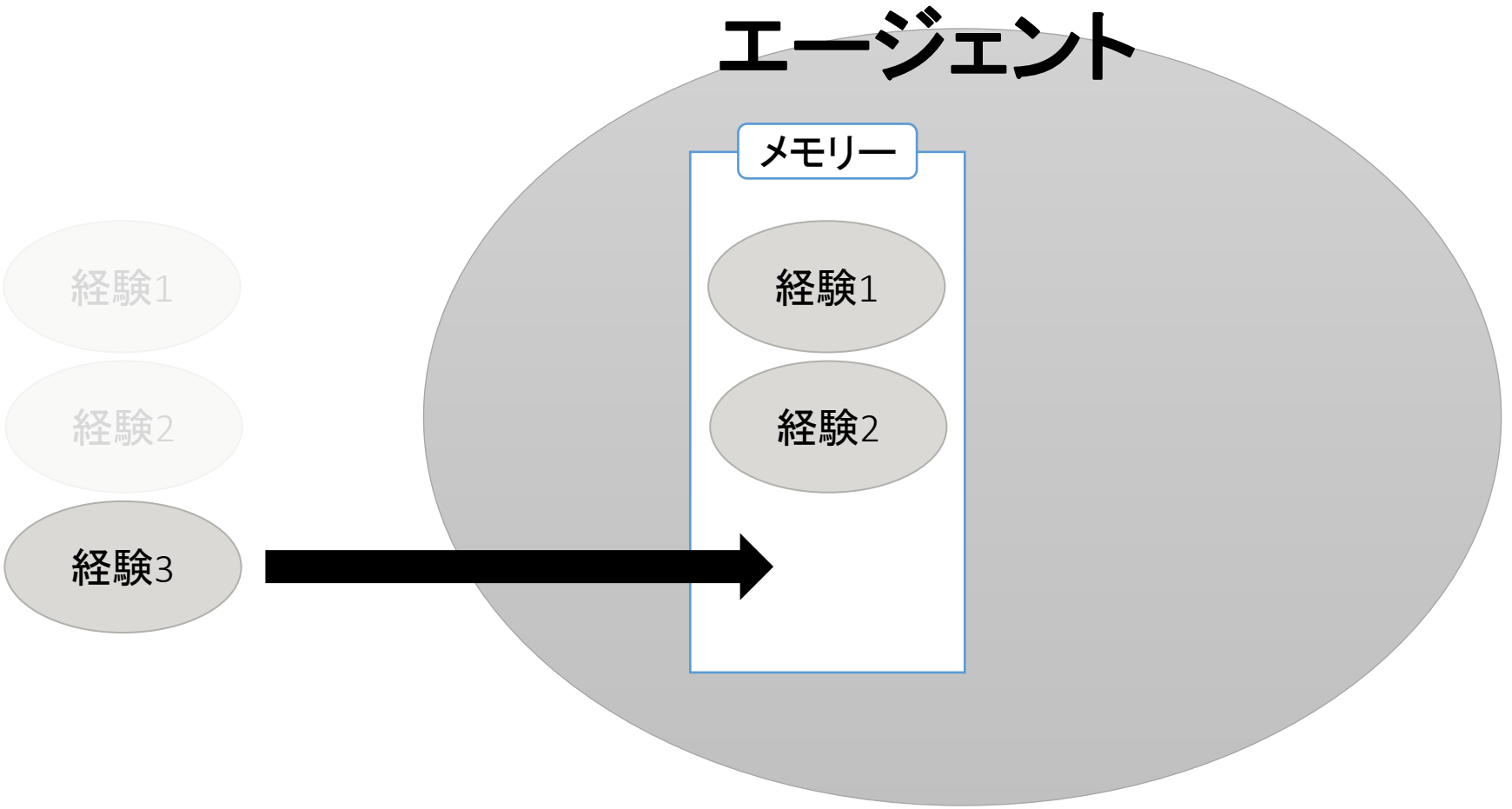
Experience Replayのイメージ

- 経験(状態、行動、報酬、遷移)をメモリーに蓄積し、メモリーからランダムサンプリングして学習する



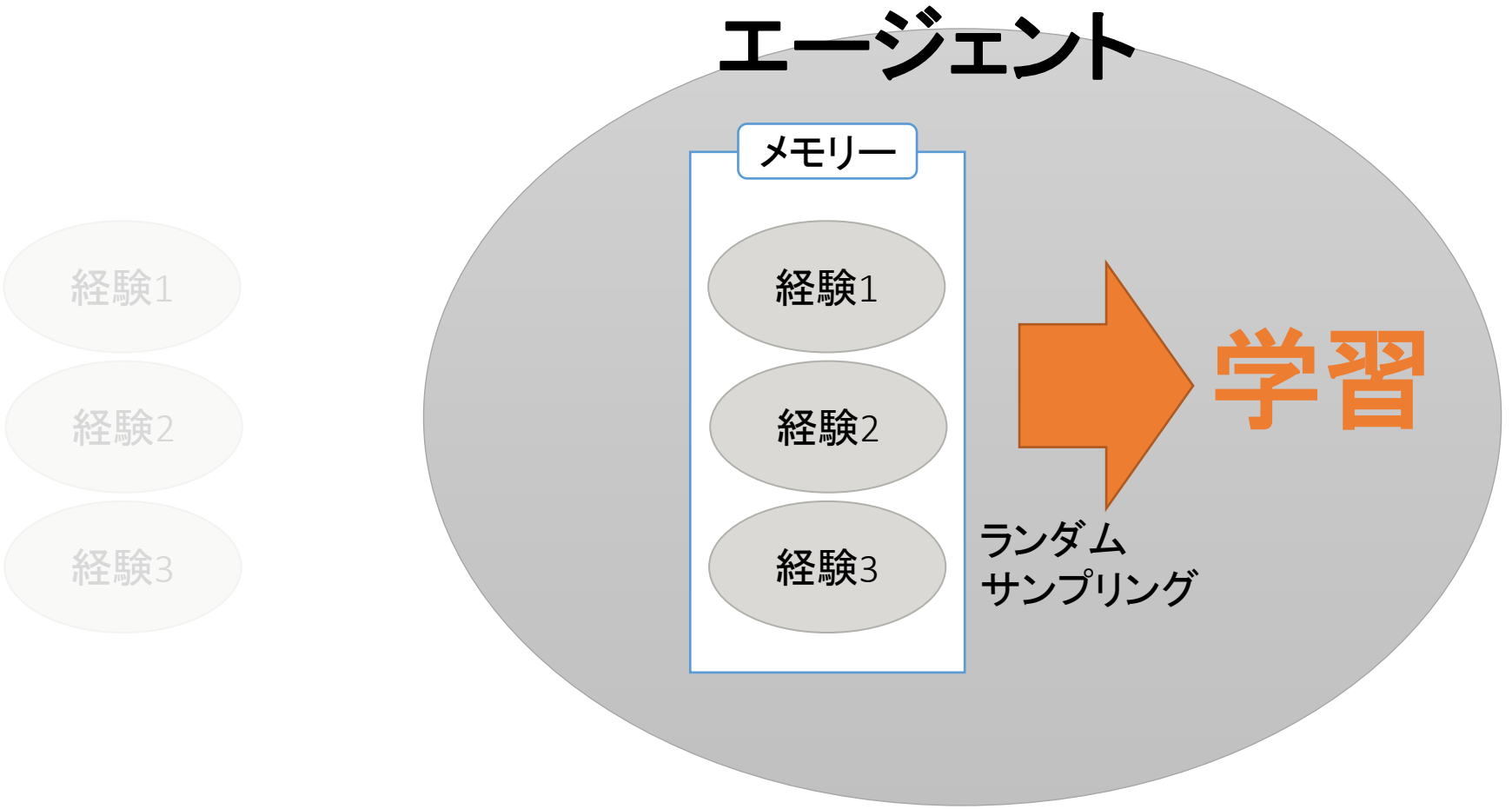
Experience Replayのイメージ

- 経験(状態、行動、報酬、遷移)をメモリーに蓄積し、メモリーからランダムサンプリングして学習する



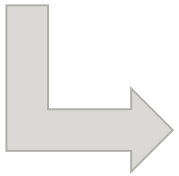
Experience Replayのイメージ

- 経験(状態、行動、報酬、遷移)をメモリーに蓄積し、メモリーからランダムサンプリングして学習する



Experience Replayで何がうれしいか?

- バッチ学習となっている



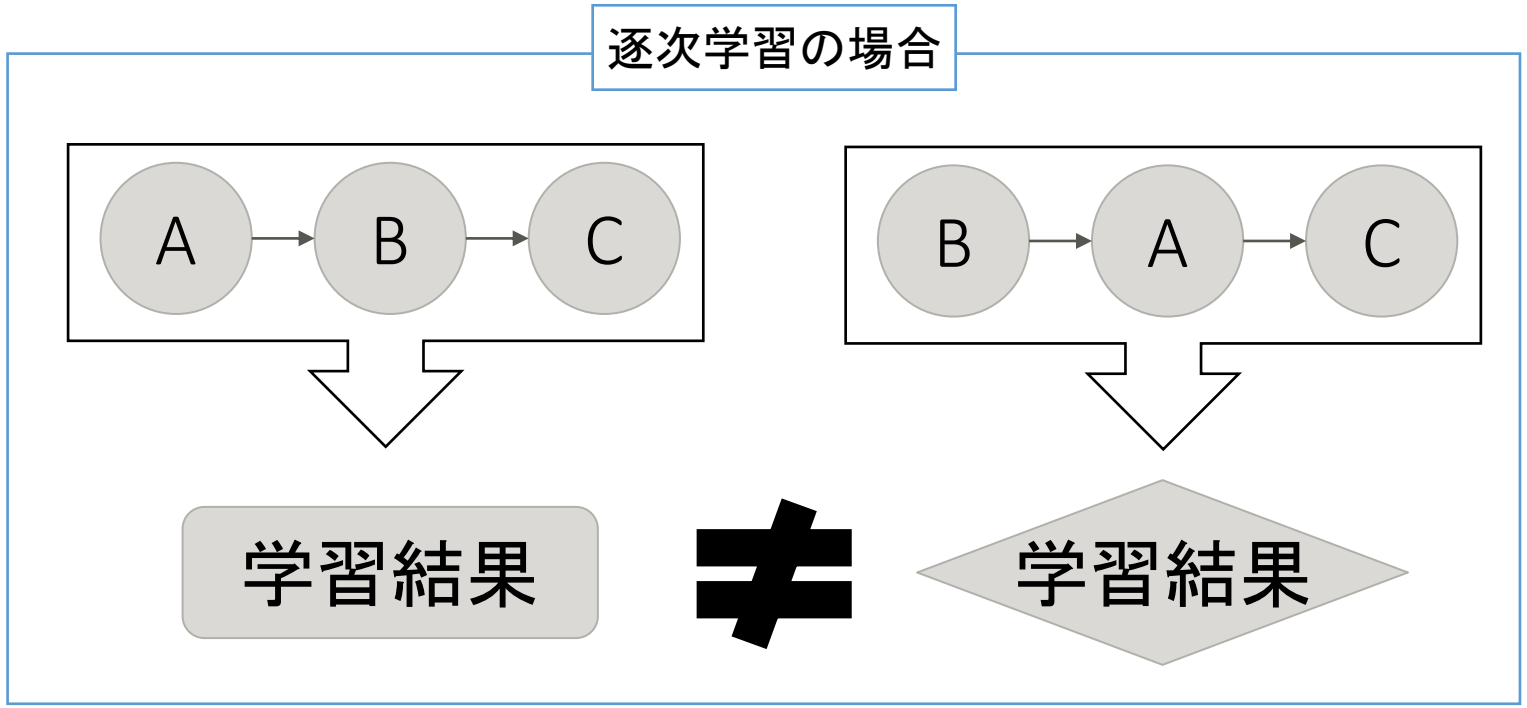
経験する順番の影響が小さく

安定した学習が期待できる

Experience Replayで何がうれしいか?

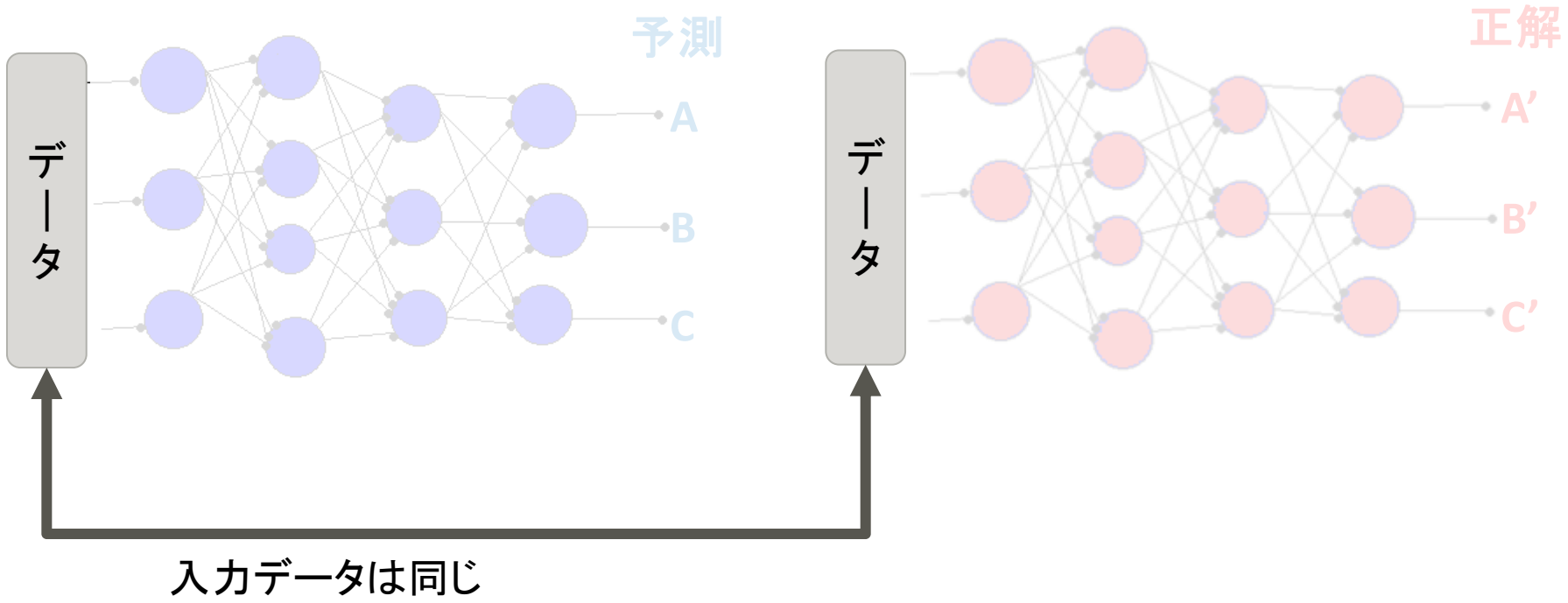
- バッチ学習となっている

  経験する順番の影響が小さく
  安定した学習が期待できる



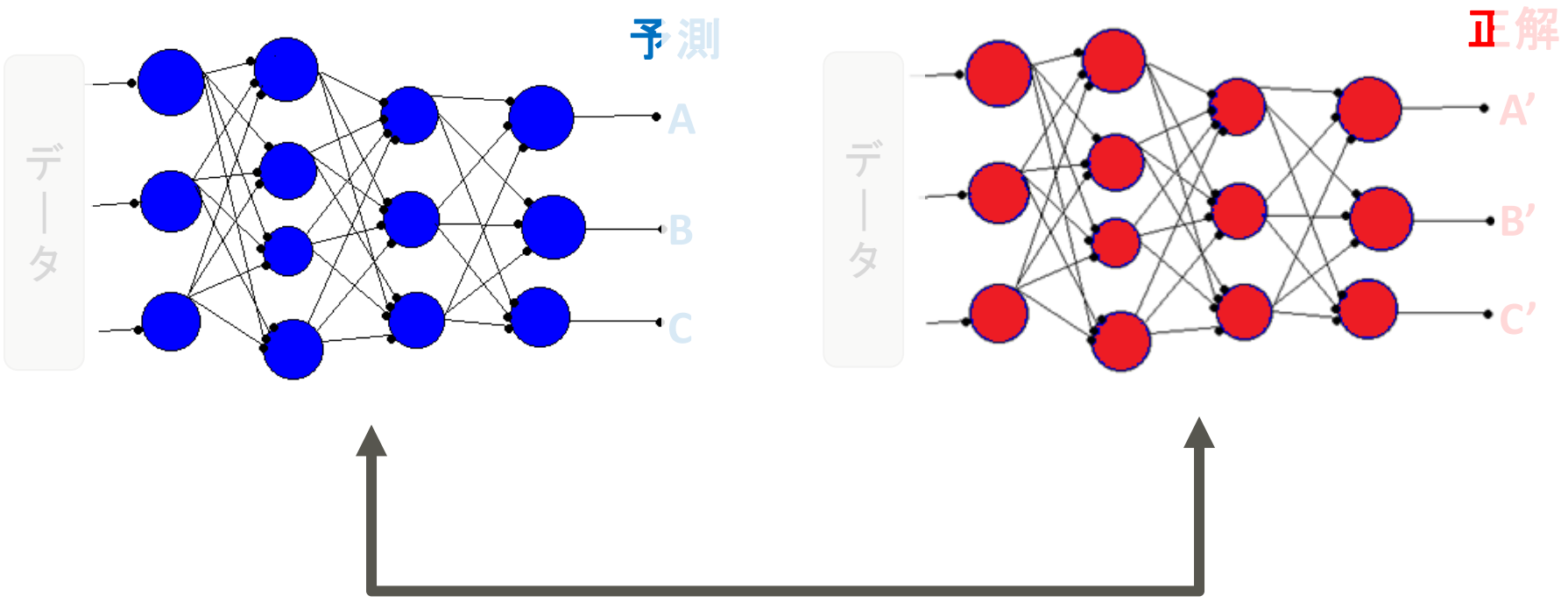
Target Q-Networkのイメージ

- 予測のQ関数(ネットワーク)と目的のQ関数(ネットワーク)を別々に作り、一定間隔で同期する。



Target Q-Networkのイメージ

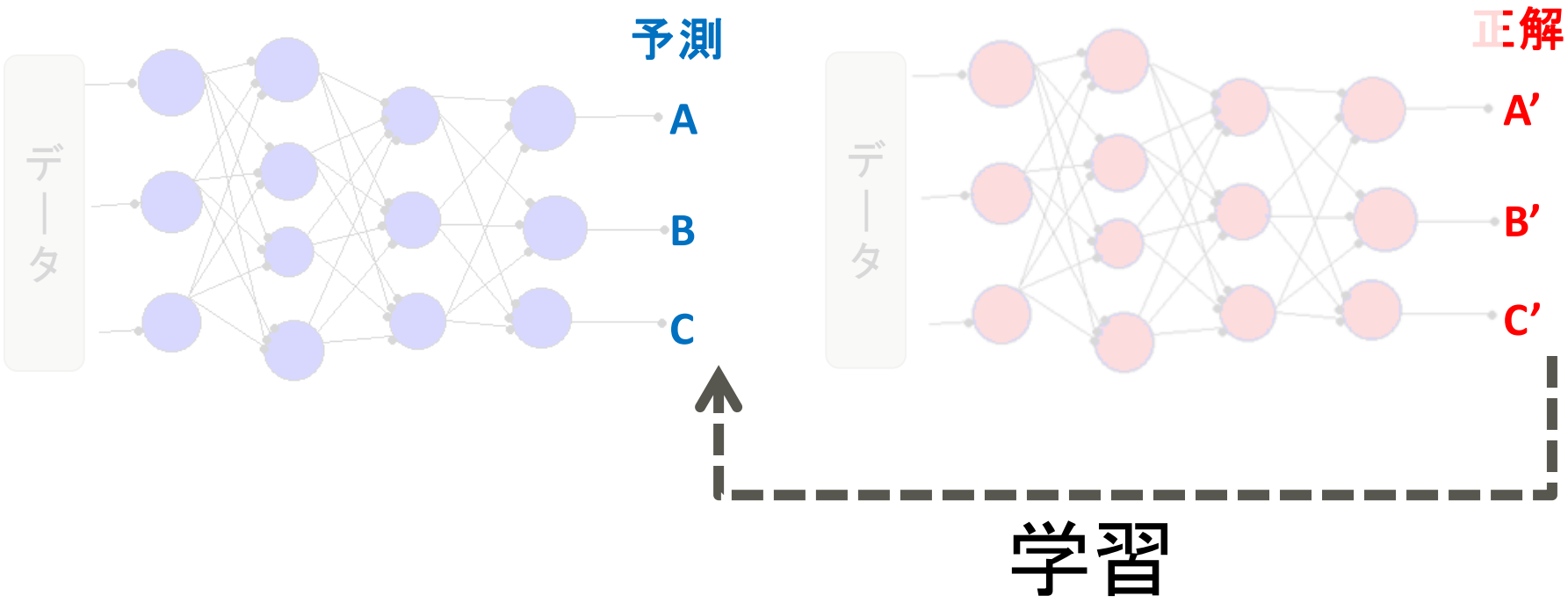
- 予測のQ関数(ネットワーク)と目的のQ関数(ネットワーク)を別々に作り、一定間隔で同期する。



別々のニューラルネットワーク
ただし一定周期で同期する

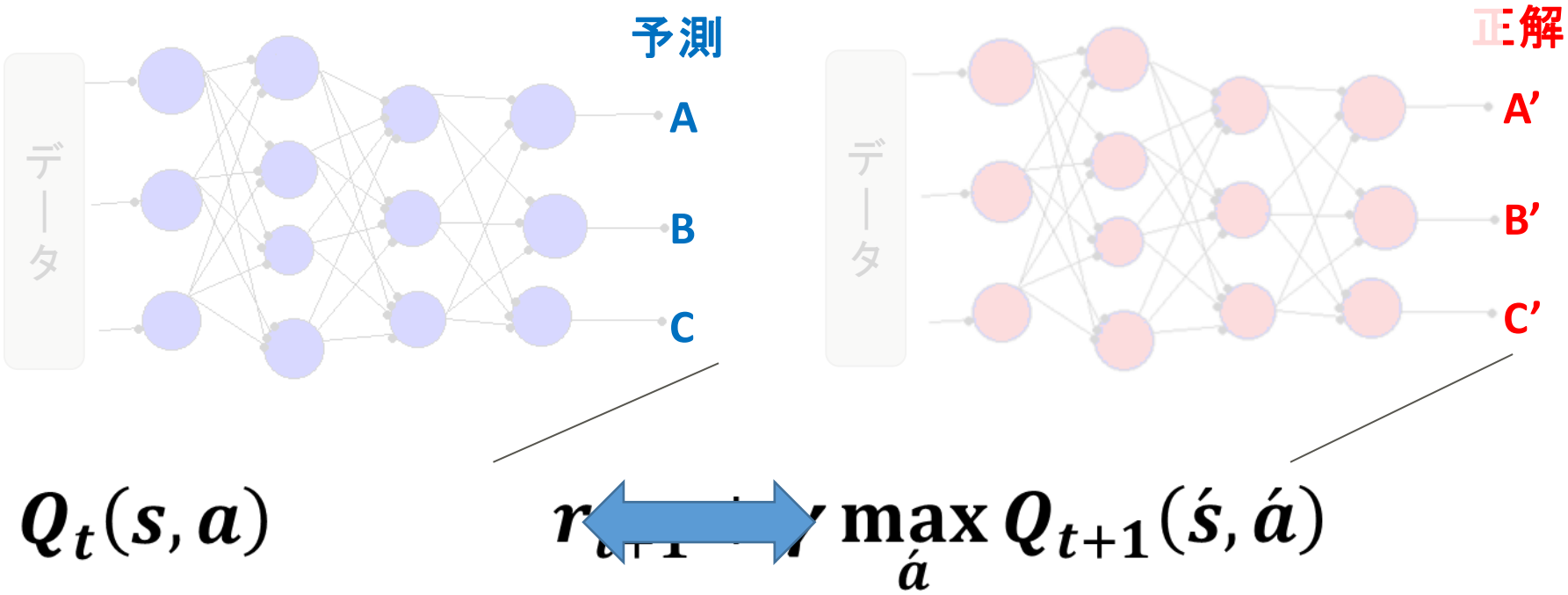
Target Q-Networkのイメージ

- 予測のQ関数(ネットワーク)と目的のQ関数(ネットワーク)を別々に作り、一定間隔で同期する。



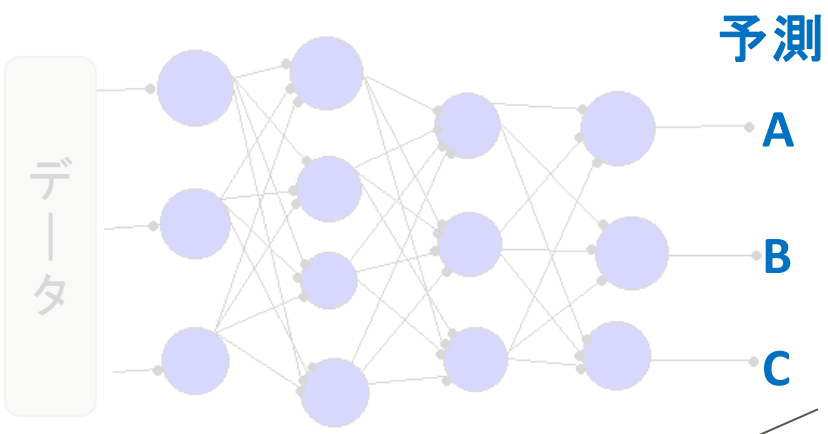
なぜTarget Q-Networkが必要か

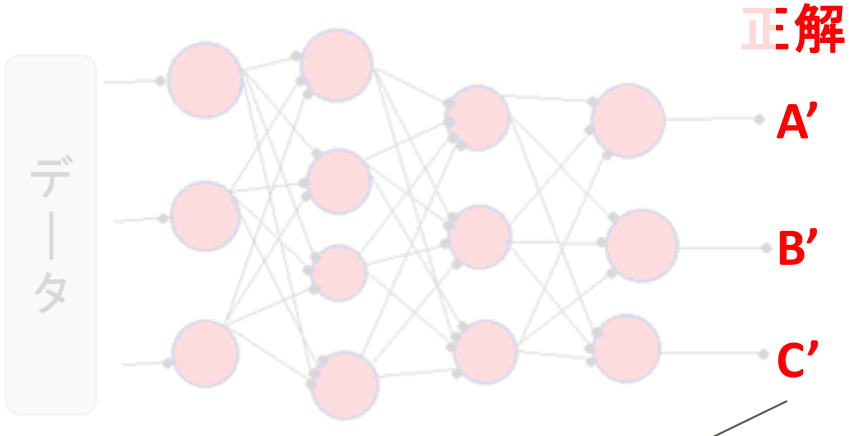
- 1つのネットワークで学習を行った場合、価値関数の更新で方策がコロコロ変わってしまう



なぜTarget Q-Networkが必要か

- 1つのネットワークで学習を行った場合、価値関数の更新で方策がコロコロ変わってしまう





$$Q_t(s,a) \quad r_{t+1} + \gamma \max_{\acute{a}} Q_{t+1}(\acute{s},\acute{a})$$

1つ前の重みに依存

Clipping

- 報酬のClipping
  - Atariなどのゲームは種類によって報酬(スコアのレンジ)が+1や+20などまちまちなので、報酬が負だったら-1、正だったら1、0はそのままにする。
  - 汎用性が高くなる。
  - 報酬の大小を区別できなくなるデメリットもある。

よくある勘違い

- エージェントは環境の変化に対応する
  - 経験の範囲で環境の変化に対応する(⇒十分な経験が必要)

- Q値=報酬
  - 報酬は状態(結果)の良さ、Q値はとる行動の良さ

- Experience Replayは過去の経験を繰り返し学習することで少ない経験でも高い精度を得ることができる工夫である
  - バッチ学習による安定した学習のため