強化学習

浅川伸一[1]

2019年11月06日

[1]東京女子大学 asakawa@ieee.org

Demo files in colab

- ランダム探索 

- 方策勾配法による迷路探索 

- SARSAによる迷路探索 

- Q学習による迷路探索 

複雑な状況をどう理解して解決するのか？

- 強化学習というニューラルネットワークモデルがあるわけではない
- 動的で複雑な環境に対処 → 強化学習 + DL → 一般人工知能への礎
- DQN ATARIのビデオゲーム,
  https://www.nature.com/articles/nature14236
- AlphaGo 囲碁,
  https://www.nature.com/articles/nature16961
- AlphaGoZero 囲碁,
  https://www.nature.com/articles/nature24270

Deep Q Network

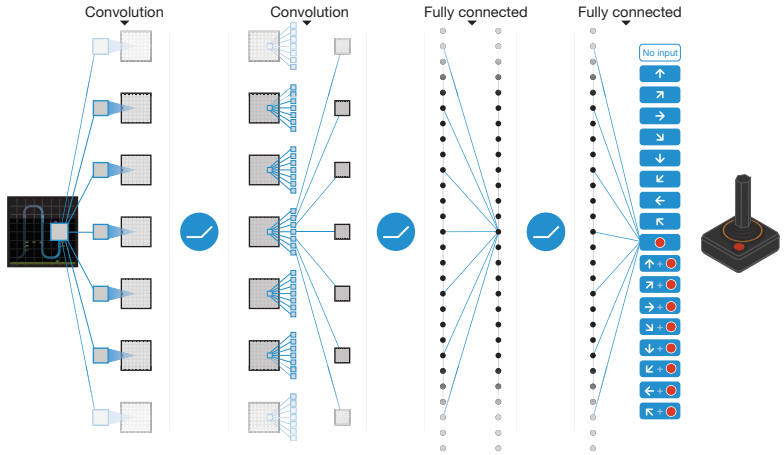


Figure 1: [4] より

- **Q 学習** Q learning に DNN を採用
- CNN が LeNet，[3] そうであったように，強化学習 RL も昔からの技術 [5]
- ではなぜ，今になって囲碁や自動運転に応用できるようになったのか？
  - ⇒コンピュータの能力，データ規模，アルゴリズムの改良，エコシステム(ArXiv, Linux, Git, ROS, AMT, TensorFlow)

強化学習

- 強化学習 $\Rightarrow$ 意思決定
  - エージェント agent が 行動(行為) action をする
  - 行動によって 状態 が変化する
  - 環境 から与えられる 報酬 によって目標が決定
- 深層学習: $\Rightarrow$ 表現，表象
  - 教師信号として目標が与えられる
  - 目標を達成するために外部状況の 表現 を獲得

強化学習 $+$ 深層学習 $=$ 人工知能

- 強化学習 $\Rightarrow$ 目標の設定
- 深層学習 $\Rightarrow$ 内部表象の獲得機構を提供

用語の整理

- 教師信号なし 報酬信号 reward signal
- 遅延フィードバック
- 価値 Value
- 行為 Action
- 状態 State
- TD 学習
  - **Sarsa**
  - **Q 学習**
  - アクタークリティック
- 報酬 R_t スカラ値
  - 時刻 tでエージェントのとった行動を評価する指標
  - エージェントは累積報酬 cumulative reward の最大化する
  - 報酬仮説: 目標は累積期待報酬の最大化として記述可能

デモ

- ブロック崩し:
  https://www.youtube.com/watch?v=V1eYniJ0Rnk
- スペースインベーダー:
  https://www.youtube.com/watch?v=W2CAghUiofY
- OpenMind selfplay:
  https://www.youtube.com/watch?v=0Bcjhp4KSgQ

教科書，参考文献など

- An Introduction to Reinforcement Learning, Sutton and Barto, 1998, MIT Press, 1998
  - http://incompleteideas.net/book/the-book.html, [5]
  - 翻訳強化学習 https://www.amazon.co.jp/dp/4627826613
- Algorithms for Reinforcement Learning, Szepesvari, Morgan and Claypool, 2010 https://sites.ualberta.ca/~szepesva/papers/RLAlgsInMDPs.pdf
- デービッド・シルバーの講義 http://www0.cs.ucl.ac.uk/staff/d.silver/web/Teaching.html
- ジョン・シュルマンのビデオ講義 https://www.youtube.com/watch?v=oPGVsoBonLM
- これからの強化学習 https://www.amazon.co.jp/dp/4627880316/

DQN の動画

- https://www.youtube.com/watch?v=TmPfTpjtdgg
- https://www.youtube.com/watch?v=W2CAghUiofY

DQN 結果

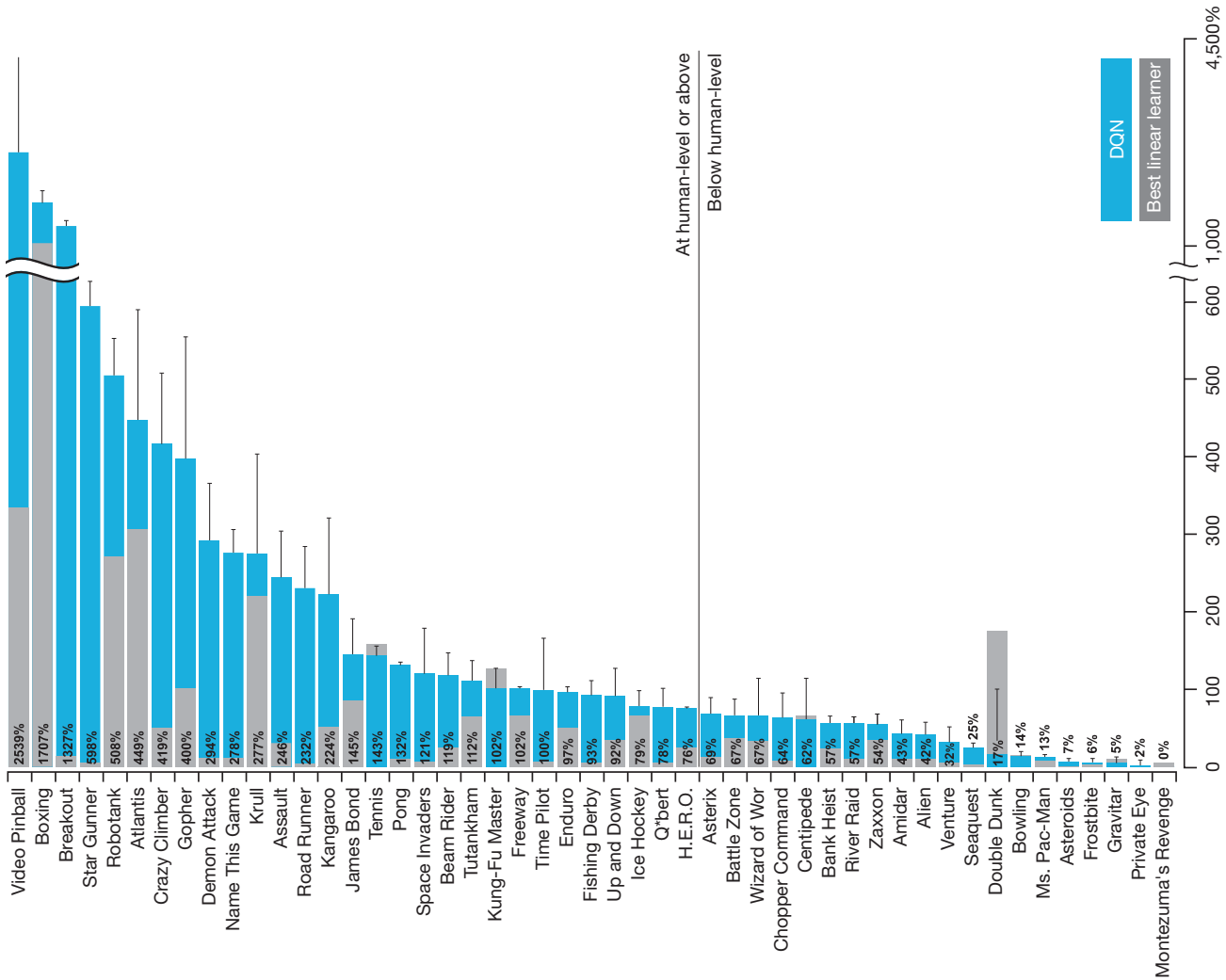


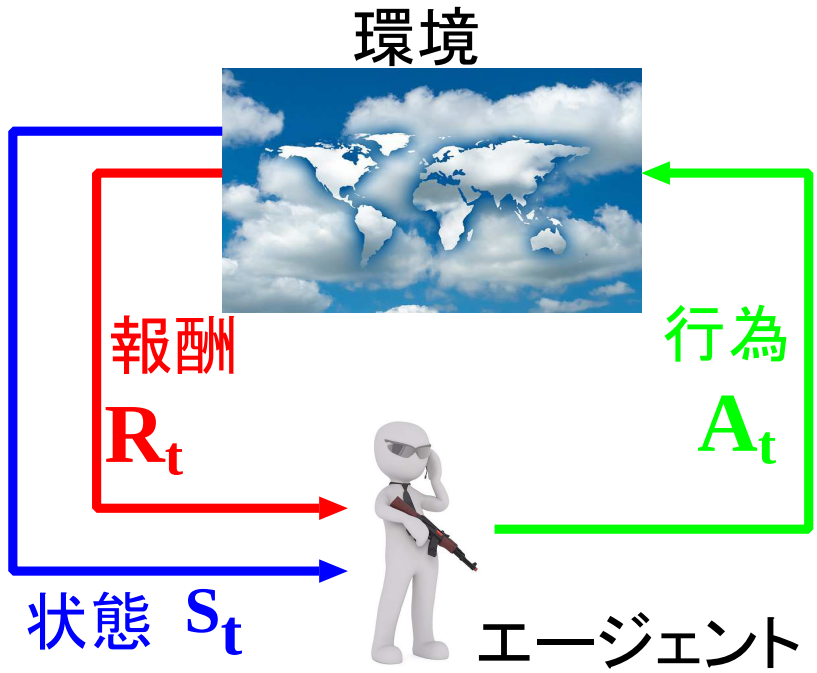
Figure 2: From [4]

なぜ DQN には難しいのか？

- Montenzuma's Revenge の動画
  https://www.youtube.com/watch?v=Klxxg9JM5tY
- Private Eyes の動画
  https://www.youtube.com/watch?v=OfyS-Wj1M78

人間にはできて強化学習には難しいこと [2]

エージェントと環境



画像出典: https://pixabay.com/en/globe-clouds-sky-background-earth-3382522/, https://pixabay.com/en/weapon-guard-soldier-protection-1816313/

- エージェント: 学習と意思決定を行う主体
  1. 行動 action A_t を行い
  2. 環境の 観察 observation O_t を行う
  3. 環境からスカラ値の 報酬 reward R_t を受け取る
- 環境: エージェント外部の全て
  1. エージェントから 行為 A_t を受け取り
  2. エージェントに 観察 O_{t+1} を与え
  3. エージェントへ 報酬 R_{t+1} を与える

エージェントの要素

- 方策 Policy
- 価値関数 Value function
- モデル エージェントが持つ環境の表象

方策 policy

- 方策：エージェントの行為
- 決定論的方策: $a = \pi(S)$
- 確率論的方策: $\pi(a|s) = p(A_t = a|S_t = s)$

価値関数

- 将来の報酬予測
- 状態評価(良/悪)
- 行為の選択

$$v_\pi(S) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s\right] \qquad (1)$$

強化学習のモデル

- 価値ベース
  - 方策:なし
  - 価値関数:あり
- 方策ベース
  - 方策:あり
  - 価値関数:なし
- アクター=クリティック Actor Critic
  - 方策: あり
  - 価値関数: あり
- モデルフリー
  - 方策，価値関数: あり
  - モデル: なし
- モデルベース
  - 方策，価値関数: あり
  - モデル: あり

探索と利用のジレンマ Exploration and exploitaion dilemma

- 過去の経験から，一番良いと思う行動ばかりをしていると，さらに良い選択肢を見つけ出すことができない 探索不足
- 更に良い選択肢ばかり探していると過去の経験が活かせない 過去の経験の利用不足

目標，収益，報酬

- エージェントの目標は累積報酬を最大化すること (報酬仮説)
  - 報酬仮説 Reward Hypothesis
  - 目標: 期待報酬の最大化
- 時刻 t における報酬 R_t: スカラ値
- 時刻 t におけるエージェント行為の評価

逐次的意思決定 Sequential Decision Making

- 目標 Goal: 総収益を最大化する行動を選択すること
- 行為，行動 Actions は長期的結果
- 収益は遅延することも有る
- 直近の報酬を選ぶよりも，長期的な報酬を考えた方が良い場合がある

収益 Return

- 収益 return G_t: 割引付き収益

$$G_t = R_{t+1} + \gamma R_{t+2} + \ldots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2}$$

- 割引率 The discount $\gamma \in [0, 1]$: 現時点から見た将来の報酬を計算するため
  - 遅延報酬 delayed reward の評価
  - 0 に近ければ 近視眼的 評価
  - 1 に近ければ 将来を見通した 評価

価値関数 Value Function

- 状態価値関数 v と 行動価値関数 q
- 状態価値関数 state-value function:

$$v_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s] \tag{3}$$

$$v_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma v_\pi(S_{t+1} | S_t = s)] \tag{4}$$

- 行動価値関数 action-value function:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a] \tag{5}$$

$$q_\pi(s, a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{6}$$

最適価値関数 Optimal Value Function

- 最適状態価値関数:

$$v_* (s) = \max_\pi v_\pi (s)$$

- 最適行動価値関数:

$$q_* (s, a) = \max q_\pi (s, a)$$

- ベルマン方程式一般に非線形になるので難しい

最適価値関数 Optimal Value Functions

- 最大の価値を与える関数

$$Q^{*}(s,a) = \max_{\pi} Q^{\pi}(s,a) = Q^{\pi^{*}}(s,a)$$

- 最適価値関数 Q^{*} が得られれば最適方策 π^{*} を求めることができる

$$\pi^{*}(s) = \mathrm{argmax}_{a} Q^{*}(s,a)$$

- 全ての意思決定における最適価値:

$$\begin{aligned} Q^{*}(s,a) &= r_{t+1} + \gamma \max_{a_{t+1}} r_{t+2} + \gamma^{2} \max_{a_{t+2}} r_{t+3} + \ldots \\ &= r_{t+1} + \gamma \max_{a_{t+1}} Q^{*}(s_{t+1}, a_{t+1}) \end{aligned}$$

- ベルマン方程式 Bellman equation:

$$Q^{*}(s,a) = \mathbb{E}_{s'}\left[r + \gamma \max_{a'} Q^{*}(s', a') \,|s, a \right].$$

マルコフ状態

$$P\left(S_{t+1}\,|S_t\right) = P\left(S_{t+1}\,|S_1, \ldots, S_t\right)$$

- 未来と過去とは無関係

$$H_{1:t} \to S_t \to H_{t+1:\infty}$$

H: 履歴

- 一度状態 S が決まれば過去の履歴は不要
- 逆に言えば状態 S は未来に対する十分統計量
- 環境の状態 S_t^e はマルコフ性を持つ
- 歴史 H_t はマルコフ性を持つ

完全観測可能，部分観測可能 Full, partially observability

$$O_t = S_t^a = S_t^e$$

- 部分観測可能なマルコフ決定過程 **POMDP**: **P**artially **O**bservable **M**arkov **D**ecision **P**rocess
- マルコフ決定過程 Markov decision processes: MDP
  - ほぼ全ての強化学習はマルコフ決定過程として記述可能

マルコフ過程 Markov Process

- マルコフ過程 MP は
- マルコフ過程 Markov Process (マルコフ連鎖 Markov Chain): 状態 S と遷移行列 P
  - S: 状態の集合
  - P: 状態間の遷移行列
  - $P_{ss'} = P\left(S_{t+1} = s' \,|S_t = s\right)$

マルコフ過程決定過程 Markov Decision Process: MDP

- MDP は，状態 S, 行動 A, 遷移確率 P, 報酬 R の組 $\langle S, A, P, R, \gamma \rangle$

$$P^a_{ss'} = P\left(S_{t+1} = s' \,|S_t = s, A_t = a\right)$$

- R は報酬関数 $R^a_s = \mathbb{E}\left[R_{t+1} \,|S_t = s, A_t = a\right]$
- γ は割引率 discount factor $\gamma \in [0, 1]$

グリッドワールド

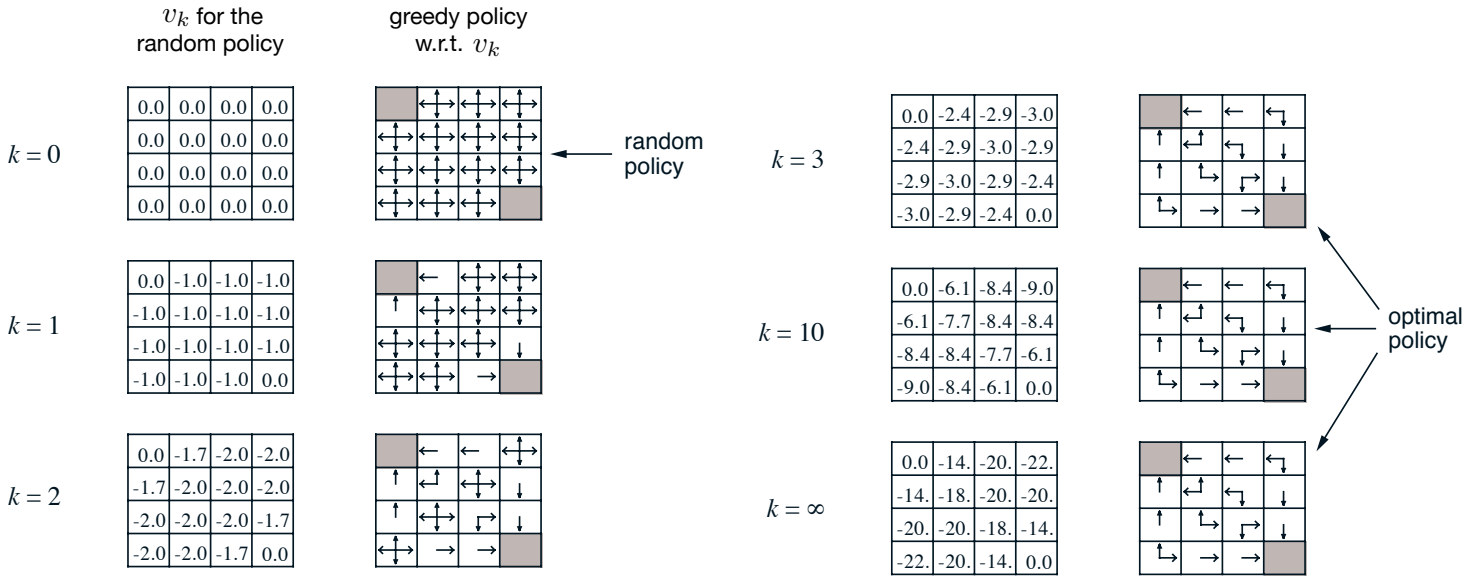


Figure 3: From [5]

価値反復 Value Iteration と方策反復 Policy Iteration

Policy Iteration (using iterative policy evaluation) for estimating $\pi \approx \pi_*$

1. Initialization
   $V(s) \in \mathbb{R}$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$

2. Policy Evaluation
   Loop:
   $\Delta \leftarrow 0$
   Loop for each $s \in \mathcal{S}$:
   $v \leftarrow V(s)$
   $V(s) \leftarrow \sum_{s',r} p(s', r \mid s, \pi(s))\big[r + \gamma V(s')\big]$
   $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
   until $\Delta < \theta$ (a small positive number determining the accuracy of estimation)

3. Policy Improvement
   policy-stable $\leftarrow$ *true*
   For each $s \in \mathcal{S}$:
   old-action $\leftarrow \pi(s)$
   $\pi(s) \leftarrow \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
   If *old-action* $\neq \pi(s)$, then *policy-stable* $\leftarrow$ *false*
   If *policy-stable*, then stop and return $V \approx v_*$ and $\pi \approx \pi_*$; else go to 2

[5]

Value Iteration, for estimating $\pi \approx \pi_*$

Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
| $\Delta \leftarrow 0$
| Loop for each $s \in \mathcal{S}$:
| $v \leftarrow V(s)$
| $V(s) \leftarrow \max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$
| $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

[5]

Figure 4: 左:方策反復 PI, 右:価値反復 VI

一般解

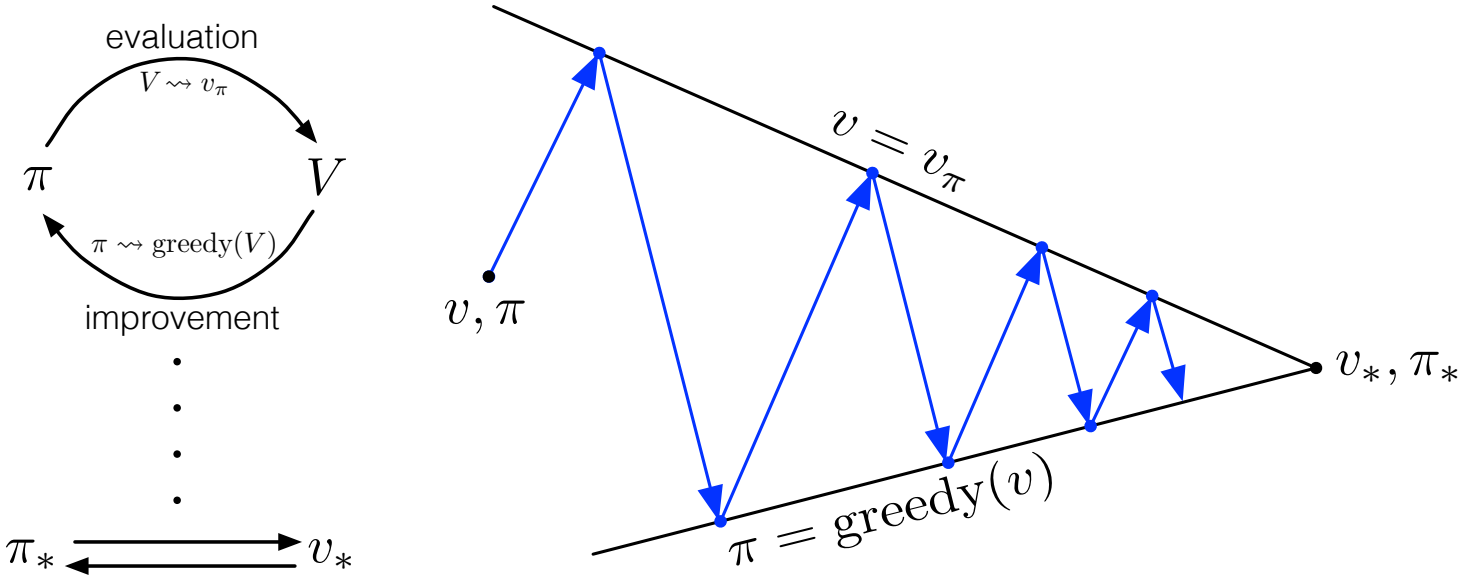


Figure 5: From [5]

$$\pi_0 \rightarrow v_{\pi_0} \rightarrow \pi_1 \rightarrow v_{\pi_1} \cdots \pi_* \rightarrow v_* \tag{7}$$

アクタークリティック

- $Q(s, a, \mathbf{w}) \sim Q^{\pi}(s, a)$
- ポリシーパラメータ u を SGD で更新

$$\frac{\partial \ell}{\partial u} = \frac{\partial \log \pi(a|s, u)}{\partial u} Q(s, a, w) \tag{8}$$

or

$$\frac{\partial \ell}{\partial u} = \frac{\partial Q(s, a, w)}{\partial a} \frac{\partial a}{\partial u} \tag{9}$$

Asynchronous Advantage Actor-Critic (A3C)

- 状態価値関数の推定

$$V(s,u) \sim \mathbb{E}\left[r_{t+1} + \gamma + r_{t+2} + \ldots | s\right] \tag{10}$$

- Q 値を推定

$$q_t = r_{t+1} + \gamma r_{t+2} + \ldots + \gamma^{n-1} r_{t+n} + \gamma^n v(s_{t+n}, v) \tag{11}$$

- アクターをターゲットに向けて更新

$$\frac{\partial \ell_u}{\partial u} = \frac{\partial \log \pi(a_t | s_t, u)}{\partial u} (q_t - V(s,t)) \tag{12}$$

- 批評家の更新

$$\ell_u = \left(q_t - V(s_t, v)^2\right) \tag{13}$$

迷宮探索 UNREAL

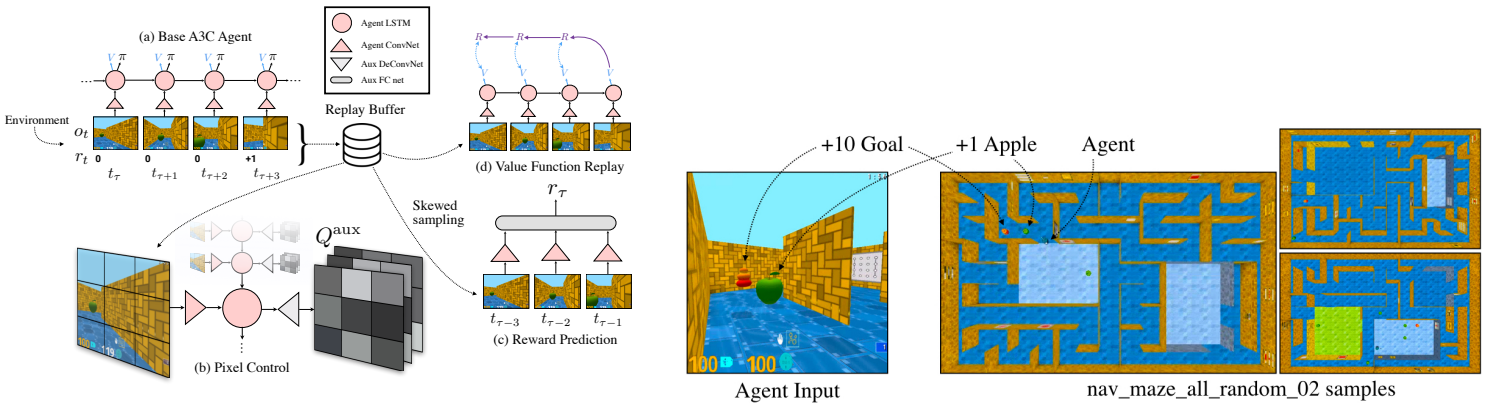


Figure 6: From [1]

A3C アルゴリズム

- ソフトマックス関数によるポリシー $\pi(a|s_t)$ のエンドツーエンド学習
- 時刻 t における観測 o_t は画面の画素データ
- 状態 $s_t = f(o_1, \ldots, o_t)$ はリカレントニューラルネットワーク(LSTM)
- 価値関数 $V(s)$ とすべての行為での $\pi(a|s)$
- Task is to collect apples ($+1$ reward) and escape ($+10$ reward)

[1]Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., & Kavukcuoglu, K. (2016). Reinforcement learning with unsupervised auxiliary tasks. *arXiv*.

[2]Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2017). Building machines that learn and think like people. *Behavioral and Brain Sciences*, 1–72.

[3]LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324.

[4]Mnih, V., Kavukchuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstr, D., Legg, S., & Hassbis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*, 529–533.

[5]Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.