確率的傾斜法を用いた強化学習とロボットへの適用

非会員　木　村　　元
東京工業大学 大学院総合理工学研究科
非会員　小　林　重　信
東京工業大学 大学院総合理工学研究科

キーワード：
強化学習，ロボット

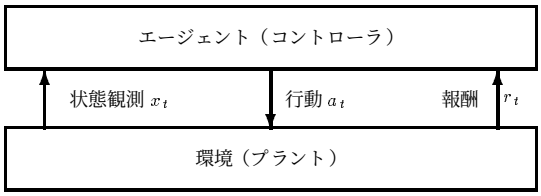

図 1　強化学習の枠組。エージェントは試行錯誤を通じて適切な制御規則を獲得していく。

1. はじめに

強化学習は，試行錯誤を通じて制御規則を自動的に獲得する学習制御の枠組として有望である。本稿は強化学習アルゴリズムの１つである確率的傾斜法をロボットへ適用した例について紹介し，強化学習のメリットについて述べる。

2. 強化学習の枠組とその特徴

強化学習の枠組を図 1に示す。エージェントは，利得（return: 最も単純な場合，報酬の総計）の最大化を目的として，状態観測から行動出力へのマッピング（政策（policy）と呼ばれる）を獲得する。環境やエージェントには一般に下記の性質が想定される。

- エージェントはあらかじめ環境に関する知識を持たない
- 環境の状態遷移は確率的
- 報酬の与えられ方は確率的
- 状態遷移を繰返した後，やっと報酬にたどり着くような，段取り的な行動を必要とする環境（報酬の遅れ）

強化学習が注目を集める理由の一つは，不確実性のある環境を扱っている点にある。多くの実世界の制御問題において，厄介な不確実性の扱いが求められる。もう一つの理由は，報酬に遅れが存在し，離散的な状態遷移も含んだ段取り的な制御規則の獲得を行う点にある。設計者がゴール状態で報酬を与えるという形で，させたいタスクをエージェントに指示しておけば，ゴールへの到達方法はエージェントの試行錯誤学習によって自動的に獲得される。つまり，設計者は「何をすべきか」をエージェントに報酬という形で指示しておくだけで「どのように実現するか」をエージェントが学習によって自動的に獲得する枠組となっている。応用上，強化学習を用いることにより以下のメリットがあると考えられる。

- 十分に優れた性能を持つ強化学習エージェントをコントローラとして１つだけ開発しておけば，あとはロボットの目的に応じて報酬の与え方だけを設計者が設定するだけで，あらゆる種類のロボット制御方法を同一のコントローラによって自動的に獲得できる。
- 試行錯誤を通じて学習するため，人間のエキスパートが得た解よりも優れた制御方法を発見する可能性がある。特に環境に不確実な要素（摩擦やガタ，振動や誤差など）が多い場合，人間の常識はあてにならないので，強化学習の効果が期待できる。エキスパートの制御規則を学習初期状態に設定して，それを改善する場合と，全くのゼロから学習を開始し，設計者にとっては意外な新しい解を発見する場合とが考えられる。また，プラント特性の経年劣化等の環境変化に対して自動的に追従することも期待できる。

強化学習の詳細な理論的解説は，文献(1)~(4) (6)~(9)を参照。

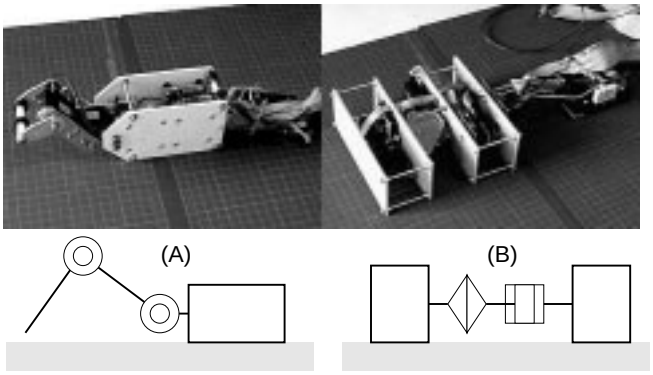

図 2　学習対象としたロボット機構とその模式図。A はボディから２節リンクアームが張り出す構造を持ち，B はボディにねじりと曲げを行う構造を持つ。各ロボット後方（写真右側）に見えるのは移動距離計測器である。A と B はメカニズム的に全く異なるが，両方へ完全に同じ学習アルゴリズムの適用を試みる。

3. ロボットの歩行制御規則の獲得問題

図 2に示すように，モータを２個搭載した２自由度の機

Reinforcement Learning using Stochastic Gradient Algorithm and its Application to Robots

By *Kimura Hajime*, Non-member and *Kobayashi Shigenobu*, Non-member (Interdisciplinary Graduate School of Sci. and Eng., Tokyo Institute of Technology)

構を持つロボット A および B に対し，完全に同一の強化学習アルゴリズムを適用し，効率よく前進する動作の獲得を試みる。エージェントが獲得すべき制御規則は，現在の関節の角度を状態入力として与えられたとき，前進するような動きとなるようにモータの目標値とすべき関節の角度を出力することである。ロボットの学習目標は，効率よく前進することなので，各時刻におけるボディの前進速度をエージェントが報酬として受け取るよう設定する。エージェントとロボットは以下のやりとりを行う。

（1） エージェントは状態観測としてロボットの関節の角度θ_1, θ_2を受け取る

（2） エージェントは行動出力として関節モータの角度の目標値 a_1, a_2を出力

（3） ロボットは目標角度の方向へ各モータを動かす。

（4） 約 0.2 秒後，ロボットはボディが移動した距離を計測し，その値を報酬としてエージェントに与える。

（5） ステップ 1 に戻って繰り返す。

上記のように設定することにより，ロボットを効率よく前進させる学習問題は，エージェントが利得（報酬の総計）を最大化するよう政策を探索する最適化問題へ帰着される。

ここで注目すべき点は，ロボット A と B がメカニズム的に全く異なるにもかかわらず，強化学習問題として見ると同じ問題になることである。よって，ロボット A へ適用可能な強化学習アルゴリズムは，何も変更することなくロボット B にも適用可能である。

状態観測である関節の角度θ_1, θ_2および行動出力である関節モータの角度の目標値 a_1, a_2は，それぞれ 0 から 255 までの整数値をとる。報酬の値は-128〜127 の範囲の整数値をとり，ボディが移動しない場合は 0 である。

関節を駆動するモータとして模型用のサーボモータを用いたため，本実験における状態観測θ_1, θ_2は，直前のステップにおいてエージェントが出力した行動 a_1, a_2に等しい値とした。よって，エージェントが毎ステップにおいて直前のステップで出力した値とは大きく異なる値（値の差がおおむね 80 以上）を出力した場合，モータの応答が追い付かないため，エージェントが観測する角度状態とロボットの真の角度とが食い違う場合がある。これは隠れ状態問題 [4] と呼ばれる。

本実験では上記のような観測入力–行動出力に対応可能で，さらに隠れ状態問題も対処可能な確率的傾斜法に基づく Actor-Critic アルゴリズム [5] を用いた。

4. 強化学習アルゴリズムの実装

本実験では確率的傾斜法に基づく Actor-Critic アルゴリズム（図 3）を用いる。このアルゴリズムは，行動選択の確率分布を規定する確率的政策πの関数形が明示的に与えられると，アルゴリズムの一般式に当てはめることにより，処理すべき計算をほぼ一意に導くことが可能である。本実験では以下のような処理によって学習が行われるが，これらの式は文献 (5) の実装を参考にすると簡単に導出できる。

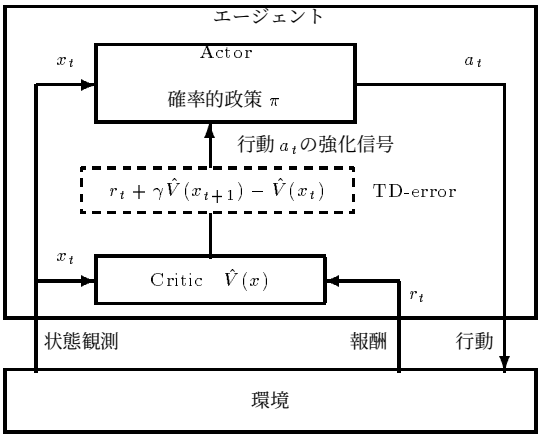

図 3 Actor-Critic アルゴリズムの構成

【エージェントの学習手順】

（1） 初期化：エージェントは actor 用の記憶変数 w_1〜w_6，$w1_{xy}, w2_{xy}$（x, yはそれぞれ 1〜6 の整数）およびそれぞれに対応した記憶変数 Trace_1〜Trace_6，$\mathrm{Trace1}_{xy}$，$\mathrm{Trace2}_{xy}$，Critic 用の記憶変数$\hat{V}_{xy}$を用意し，値を全て 0 に初期化する。

（2） 状態観測：エージェントは環境において状態$\theta_1(t)$，$\theta_2(t)$ を観測する。

（3） 行動決定：Actor において以下の処理により行動 a_1, a_2を選択する。状態観測のレンジは 0〜255 なので，この区間を 6 等分し，分割した区間を 1〜6 の整数でラベル付けする。観測した状態$\theta_1(t), \theta_2(t)$ が含まれる区間のラベル番号をそれぞれ i, jとする。以下のμ_1，μ_2，σ_1，σ_2を計算する。

$$\mu_1 = \frac{(\theta_1 - 128)}{32} w_1 + \frac{(\theta_2 - 128)}{32} w_2 + 2 \cdot \mathrm{w1}_{ij}$$

$$\sigma_1 = \frac{1}{1 - \exp(-w_3)}$$

$$\mu_2 = \frac{(\theta_1 - 128)}{32} w_4 + \frac{(\theta_2 - 128)}{32} w_5 + 2 \cdot \mathrm{w2}_{ij}$$

$$\sigma_2 = \frac{1}{1 - \exp(-w_6)}$$

Actor は中心値μ_1，標準偏差σ_1のガウス分布 $N(\mu_1, \sigma_1)$ を用いて $a1 = 32\, N(\mu_1, \sigma_1) + 128$ として行動 a_1を選択する。同様に $a2 = 32\, N(\mu_2, \sigma_2) + 128$ として行動 a_2を選択する。

（4） TD-error の計算：Critic は報酬 r_tを受け取り，次の状態$\theta_1(t+1)$，$\theta_2(t+2)$ を観測する。このとき新たに観測した状態の含まれる区間のラベル番号を I，Jとし，以下の TD-error を計算する。

$$(\text{TD-error}) = \left[r_t + \gamma\, \hat{V}_{I,J} \right] - \hat{V}_{ij} \ ,$$

γ $(0 \le \gamma \le 1)$ は割引率，$\hat{V}$は Critic が推定した割引報酬の期待値を表す。

（5） 以下の適正度を計算する。

$$e_1 = \frac{(\theta_1 - 128)}{32}\frac{(a_1 - 128)}{32}$$

$$e_2 = \frac{(\theta_2 - 128)}{32}\frac{(a_1 - 128)}{32}$$

$$e_3 = \left(\left(\frac{a_1 - 128}{32}\right)^2 - \sigma_1^2\right)(1 - \sigma_1)$$

$$\mathrm{e1}_{xy} = \begin{cases} \frac{a_1 - 128}{16} & x = i \ \&\ y = j \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } x,y.$$

$$e_4 = \frac{(\theta_1 - 128)}{32}\frac{(a_2 - 128)}{32}$$

$$e_5 = \frac{(\theta_2 - 128)}{32}\frac{(a_2 - 128)}{32}$$

$$e_6 = \left(\left(\frac{a_2 - 128}{32}\right)^2 - \sigma_2^2\right)(1 - \sigma_2)$$

$$\mathrm{e2}_{xy} = \begin{cases} \frac{a_2 - 128}{16} & x = i \ \&\ y = j \\ 0 & \text{otherwise} \end{cases} \quad \text{for all } x,y.$$

（6） 適正度を用いて適正度の履歴を更新：

$$\mathrm{Trace}_1 \leftarrow e_1 + \gamma\,\mathrm{Trace}_1$$

$$\mathrm{Trace}_2 \leftarrow e_2 + \gamma\,\mathrm{Trace}_2$$

$$\mathrm{Trace}_3 \leftarrow e_3 + \gamma\,\mathrm{Trace}_3$$

$$\mathrm{Trace1}_{xy} \leftarrow \mathrm{e1}_{xy} + \gamma\,\mathrm{Trace1}_{xy} \text{ , for all } x,y.$$

$$\mathrm{Trace}_4 \leftarrow e_4 + \gamma\,\mathrm{Trace}_4$$

$$\mathrm{Trace}_5 \leftarrow e_5 + \gamma\,\mathrm{Trace}_5$$

$$\mathrm{Trace}_6 \leftarrow e_6 + \gamma\,\mathrm{Trace}_6$$

$$\mathrm{Trace2}_{xy} \leftarrow \mathrm{e2}_{xy} + \gamma\,\mathrm{Trace2}_{xy} \text{ , for all } x,y.$$

（7） 適正度の履歴と TD-error を用いて重み変数更新：

$$w_1 \leftarrow w_1 + \alpha_p\,\mathrm{Trace}_1\,(\text{TD-error})$$

$$w_2 \leftarrow w_2 + \alpha_p\,\mathrm{Trace}_2\,(\text{TD-error})$$

$$w_3 \leftarrow w_3 + \alpha_p\,\mathrm{Trace}_3\,(\text{TD-error})$$

$$\mathrm{w1}_{xy} \leftarrow \mathrm{w1}_{xy} + \alpha_p\,\mathrm{Trace1}_{xy}\,(\text{TD-error}) \text{ ,for all } x,y.$$

$$w_4 \leftarrow w_4 + \alpha_p\,\mathrm{Trace}_4\,(\text{TD-error})$$

$$w_5 \leftarrow w_5 + \alpha_p\,\mathrm{Trace}_5\,(\text{TD-error})$$

$$w_6 \leftarrow w_6 + \alpha_p\,\mathrm{Trace}_6\,(\text{TD-error})$$

$$\mathrm{w2}_{xy} \leftarrow \mathrm{w2}_{xy} + \alpha_p\,\mathrm{Trace2}_{xy}\,(\text{TD-error}) \text{ ,for all } x,y.$$

ただしα_pは actor の学習定数を表す。

（8） TD 法を用いて critic の value の推定値を更新：

$$\hat{V}_{ij} \leftarrow \hat{V}_{ij} + \alpha\,(\text{TD-error})$$

ただしαは学習率である。

（9） ステップ（2）から繰り返す。

5. 実験結果

本実験では割引率$\gamma = 0.9$，学習率$\alpha = 0.1$，$\alpha_p = 0.002$に設定した。移動距離を計測するために 1 回転 200 パルスのロータリーエンコーダに直径 3cm の車輪を付け，パルスの個数を報酬の絶対値，回転方向を報酬の符号として計測した。報酬 1000 パルスはおよそ 50cm の距離になる。実験は 5000 ステップ程度の学習を行った。実時間でおよそ 12 分程度である。

学習によってエージェントが得た報酬の累積，すなわちスタート位置からロボットが移動する様子を図 4に示す。A と B は全く異なるメカニズムであるにもかかわらず，どちらのロボットも順調に学習を始めている。ロボット A の 3000 ステップ付近と 4500 ステップ付近や，B の 2500 ステップ付近において，報酬の獲得ができなくなるような状況に陥っている。これは，より多くの報酬を得るように政策の改善を進めていくうち，うまくいけばかなり効率よく進めるが，ちょっとしたはずみで進めなくなるような，きわどい動きを獲得し，さらに学習を進めようとして行き過ぎたためと考えられる。しかし，しばらく後には再び前進する動きに改善している。A，B 共に 5000 ステップでは学習はまだ収束していない。

学習により獲得した動作の例を図 5に示す。ロボット B については，おおむね図 5に示した動作に類似する動作し観察されなかった。ところがロボット A は図 5に示した動作以外にも学習途中においてさまざまなパターンが見られ，常に変化が観測された。特に，アーム先端を地面に触れたまま，アームを激しく上下に動かすと同時にアーム自身も曲げたり伸ばしたりして，尺取虫のように移動する様子が観測された。これはアームを下に動かすときはアーム先端と地面との摩擦力が増すため，このときアームを曲げると前進しやすく，逆にアームを上に動かすときは摩擦力が減るため，このときアームを伸ばすとほとんど後退することがないことを利用するものである。特に本実験で用いた機構では，重心の位置の都合により常に前進し続けるような動き方も存在するようである。このような動作は実際に試行錯誤しない限り，獲得したり予測するのは困難である。

本実験で使用した学習アルゴリズムでは，報酬獲得につながるのであれば探索的なランダム動作を自動的に減らすようになっているが，実験では 5000 ステップの時点においてもランダムな動作が残った。さらに学習を続けるとどうなるのかについては別の機会に述べる。

6. おわりに

本稿では，異なるメカニズムを持つロボットの制御規則獲得問題を取り上げ，試行錯誤による学習の例を示した。達成させたいタスクを「報酬」という形式で記述して強化学習の枠組を適用することによって，全く異なるメカニズムを持つ 2 種類のロボットを同一の学習アルゴリズムによってタスクの達成方法を自動的に獲得できることを示した。

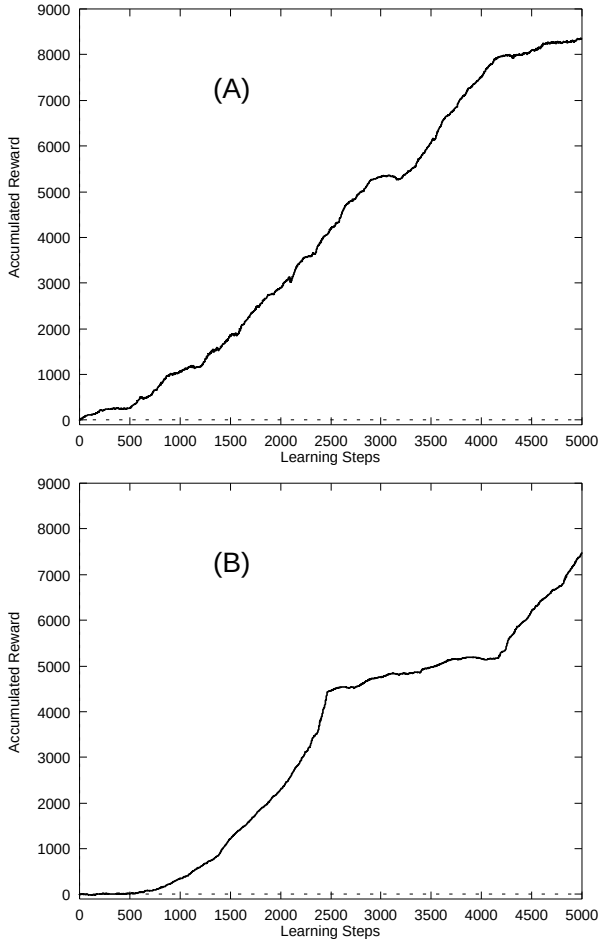

図 4　ロボット A，B の学習による移動距離の変化

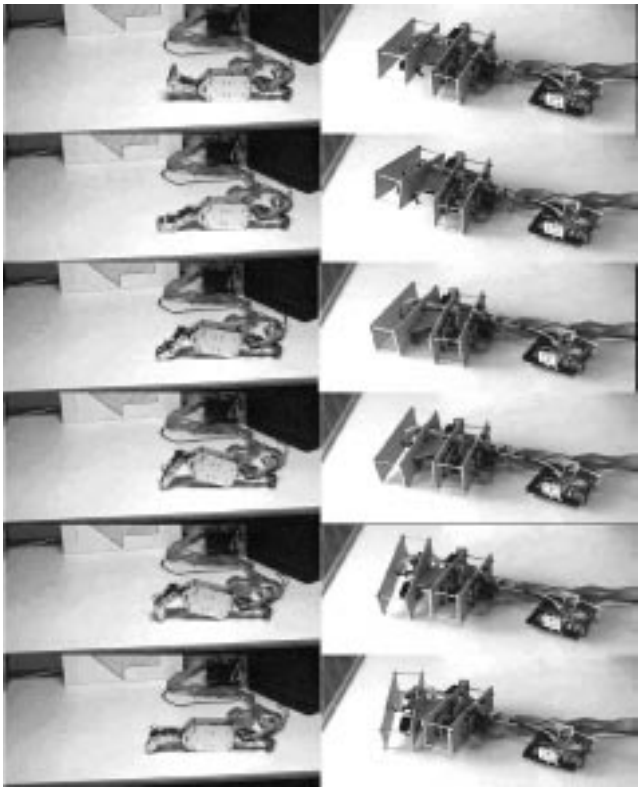

図 5　強化学習による試行錯誤を通じて得た動作例

強化学習の枠組は，ロボット以外にもさまざまな問題へ適用可能であり，今後は従来設計者が問題に応じて制御規則をプログラムしていた作業の多くを自動化していくものと期待される。

（平成年月日受付，同年月日再受付）

文　献

（1） Bertsekas, D.P. & Tsitsiklis, J. N.: *Neuro-Dynamic Programming*, Athena Scientific (1996).

（2） Kaelbling, L. P., & Littman, M. L., & Moore, A. W.: Reinforcement Learning: A Survey, *Journal of Artificial Intelligence Research*, Vol. 4, pp. 237–277 (1996).

（3） 木村 元, 山村 雅幸, 小林 重信: 部分観測マルコフ決定過程下での強化学習：確率的傾斜法による接近，人工知能学会誌, Vol.11, No.5, pp.761–768 (1996).

（4） 木村 元, Kaelbling, L. P.: 部分観測マルコフ決定過程下での強化学習，人工知能学会誌, Vol.12, No.6, pp.822–830 (1997).

（5） Kimura, H. & Kobayashi, S.: An Analysis of Actor/Critic Algorithms using Eligibility Traces: Reinforcement Learning with Imperfect Value Function, *15th International Conference on Machine Learning*, pp.278–286 (1998).

（6） 木村 元, 小林 重信: ロボットアームのほふく行動の強化学習：確率的傾斜法による接近，人工知能学会誌, Vol.14, No.1, pp.122–130 (1999).

（7） 宮崎 和光, 小林 重信: 離散マルコフ決定過程下での強化学習，人工知能学会誌, Vol.12, No.6, pp.811–821 (1997).

（8） Sutton, R. S. & Barto, A.: Reinforcement Learning: An Introduction, *A Bradford Book*, The MIT Press (1998).

（9） Watkins, C. J. C. H. and Dayan, P.: Technical Note: *Q*-Learning, *Machine Learning 8*, pp. 279-292 (1992).

木 村　元（非会員） 1992 年東京工業大学工学部制御工学科卒業。1994 年同大学大学院総合理工学研究科知能科学専攻修士課程修了。1997 年同大学大学院博士課程修了，同年 4 月日本学術振興会 PD 研究員，1998 年 4 月，東京工業大学大学院総合理工学研究科助手，現在に至る。人工知能，特に強化学習に関する研究に従事。人工知能学会，計測自動制御学会，日本ロボット学会各会員。

小 林 重 信（非会員） 1974 年東京工業大学大学院博士課程経営工学専攻修了。同年 4 月，同大学工学部制御工学科助手。1981 年 8 月，同大学大学院総合理工学研究科助教授。1990 年 8 月，教授，現在に至る。問題解決と推論制御，知識獲得と学習などの研究に従事。人工知能学会，計測自動制御学会，情報処理学会各会員。