予測を要して連続動作を含む柔軟な行動のActor-Q学習による獲得

柴田 克成*（大分大学），後藤 健太（大分大学，現在，ナブテスコ (株) 勤務）

Acquisition of Prediction-based Flexible Actions with Continuous Motions by Actor-Q-Learning

Katsunari Shibata* (Oita University)
Kenta Goto (Oita University, Presently, Nabtesco Corporation)

Abstract

In this paper, the authors first point the importance of three factors for filling the gap between humans and robots in the flexibility in the real world. Those are (1)parallel processing, (2)emergence through autonomous learning and solving "what" problems, and (3)abstraction and generalization on the abstract space. To explore the possibility of human-like flexibility in robots, a prediction-required task in which an agent (robot) gets a reward by capturing a moving target that sometimes becomes invisible was learned by Actor-Q-learning (a kind of reinforcement learning method) using a recurrent neural network. Even though the agent did not know in advance that "prediction is required" or "what information should be predicted", appropriate discrete decision making, in which 'capture' or 'move' was chosen, and also continuous motion generation in two-dimensional space could be acquired. Furthermore, in this task, the target sometimes changed its moving direction randomly when it became visible again from the invisible state. Then the agent could change its moving direction promptly and appropriately without introducing any special architecture or technique. Such emergent property is what general parallel processing systems such as Subsumption architecture do not have, and the authors believe it is a key to solve the "Frame Problem" fundamentally.

キーワード：強化学習，リカレントニューラルネット，予測，創発，フレーム問題
(Reinforcement Learning, Recurrent Neural Network, Prediction, Emergence of Intelligence, Frame Problem)

1. はじめに

「知能」という観点から，ロボットが人間との間の壁を乗り越えて次のステージに進むためには，情報であふれる実世界において人間のように柔軟かつ適切にふるまうことが求められる。そのために必要な３つの要素として，筆者らは,「並列処理」「自律学習による機能創発」「抽象化と汎化」があると考えている。

ロボットが実世界で活躍するための課題として，かつてより指摘されてきた「フレーム問題」[1] を考えてみる。ここでは，われわれがロボットの機能を開発するときに，通常,「どうやってボールの認識を実現するか」などの "How" 問題に焦点を当てるが，実世界では，まず「何を認識すべきか」といった "What" 問題を自ら解くことが重要であることを示唆し，さらにこの "What" 問題を解決する際の「逐次処理」の限界も合わせて示唆している。したがって，まずこの限界打破のために「並列処理」の導入が必須である。

また，そもそも並列ではなく逐次的なわれわれの「意識」を使ってこの "What" と "How" を同時にこなす並列処理を設計することは困難である上，トップダウンで与えると柔軟性の阻害にもなってしまう[2]。こう考えると,「何をすべきか」を直接与えないで，報酬や罰等の最低限の情報と経験を通して必要な機能が内部に勝手にでき上がってくる枠組みである「自律学習による機能創発」が必要になる。

そして，最後は「抽象化と汎化」である。われわれは将来，今と全く同じセンサ信号を得ることは恐らくない。にもかかわらず，ある状況で学習されたことは，たとえセンサ信号が違っても，目的に照らして似た状況であれば，それをうまく利用できるという素晴らしい能力を持っている。このような機能をロボット上で実現するためには，センサ信号空間を目的に沿って「抽象化」し，その抽象空間上で「汎化」を利用することが重要である。

これらの３つの条件を満たすものがニューラルネットと強化学習の組み合わせである[3]。さらに，記憶やダイナミクスを扱うためにはリカレント構造を導入したリカレントニューラルネット (RNN) が必須である。しかし，強化学習に RNN が利用されて来たものの，学習が難しいため，倒立振子[4][5][6] や離散行動の迷路問題[7][8] といった簡単なタスクへの適用に限定されてきた。最近，筆者らのグループでは，コミュニケーション[9]，文脈に基づいた行動[10]，探索[11]，過去へさかのぼってのパターンの意味の理解[12] などに有効であることを示してきた。

一方，知能ロボットやエージェントのさらなる知能化の方法として，RNN に現在のセンサ信号から未来のセンサ信号を予測することを学習させ，その内部表現を抽象的かつ文脈を考慮した状態表現として用いる方法が提案されている[13][4][14]。しかしながら，センサ信号の数が膨大になると，そのすべてを予測することはできず，予測対象をその中から設計者が選ばなければならない。つまり，前述の "What" 問題が浮上する。これに対し，著者らは，２次元平面上でしばしば不可視となる移動物体を捕獲するタスクにおいて，RNN を用いた Q 学習[17] でエージェント (ロボット) が「予測が必要である」ことも「何が予測対象か」

も明示的に与えられることなく，報酬と罰だけから予測が必要な一連の行動を学習できることを示した[15]。

本稿では，それと似た不可視物体の捕獲問題を扱う。しかし，ここでは，[15] のように1次元の離散動作ではなく2次元の連続値動作を学習する必要がある上，物体が見えない状態から再び現れる際にときどきその移動方向が変化する。物体が見えない状態で捕獲しなければならないときは，エージェントは，物体が見えなくなる前の信号から物体の位置を予測しなければならないが，もし物体が再び現れ，その動作が予測したものと矛盾している場合は，エージェントは新しい予測に基づいてその動作を柔軟に切り替えなければならない[16]。

そこでまず，2次元の連続動作の生成と「捕獲」か「移動」かの離散的な意志決定を同時に学習するために，Actor-Q 学習[18] を適用した。そして，物体の動きの矛盾が判明したときに即座に新しい予測に基づく行動に切り替えることも含めた柔軟で比較的難しい行動を，特別な手法を導入することなく，単に強化学習，つまり，報酬と罰からの学習だけで獲得できるかどうかを確認した。さらに，RNN の局所受容野の有効性に関するテストも合わせて行った。

2. ニューラルネット (NN) を用いた強化学習[3]

強化学習は試行錯誤をベースとした自律的かつ合目的的な学習であり，ニューラルネット (NN) は通常，次元の呪いによる状態の爆発を防ぐための非線形関数近似器として用いられる。上述のように，われわれのグループでは，それらの組み合わせが「並列処理」,「機能創発」,「抽象化と汎化」を可能にすると考えて来た。この学習を通して，報酬や罰だけから，報酬を得て罰を避けるために必要となる認識や記憶 (RNN を使えば) などの機能が創発する。従来，プログラムによって個々に高機能化された各機能モジュールを統合することで知能ロボットを開発しようとして来たが，ここではこの「機能モジュール」アプローチに別れを告げ，柔軟で並列な処理によって,「フレーム問題」[1] や”What” 問題を根本的に解決することを期待する。

〈2・1〉 リカレントニューラルネット (RNN) を用いた Actor-Q 学習

Actor-Q 学習[18] は，離散意志決定（ここでは行動と呼ぶ）と連続値動作生成を同時に学習するための方法である。Actor-Critic[19] を Q 学習[17] と組み合わせ， Q 値の一つを Critic として扱う。そしてこの計算を，センサ信号を入力とする一つのリカレントネット (RNN) で行う。出力は Q 出力と Actor 出力の2種類に分けられる。Q 出力は，離散意志 (行動) の決定に用いられる。動作は，Actor の出力によって決まる点を中心とする範囲内で確率的に選ばれる。つまり，時刻 t での入力 (センサ信号) を $\boldsymbol{s}_t$ とすると，Actor の出力ベクトル $\boldsymbol{m}(\boldsymbol{s}_t)$ と探索のための乱数ベクトル $\boldsymbol{rnd}_t$ の和となる。

学習については，Q 値の出力は Q 学習にしたがって，Actor の出力は Actor-Critic と同様な方法で学習するが，選択した行動の Q 値を Critic の代わりに用いる。強化学習に基づいて RNN を学習させるために，RNN の教師信号は強化学習によって求め，RNN はその教師信号を使って教師あり学習をする。選択された行動 a_t に対する Q 値の出力の教師信号は

$$Q_{train,a_t,t} = r_{t+1} + \gamma \max_a Q_a(\boldsymbol{s}_{t+1}) \cdots\cdots\cdots\cdots (1)$$

とする。ここで，r_{t+1} は時刻 $t+1$ での報酬，$Q_a(\boldsymbol{s}_{t+1})$ は時刻 $t+1$ でのセンサ信号ベクトル $\boldsymbol{s}$ が RNN の入力であるときの行動 a に対する Q 値を表す。選択されていない行動の Q 値は学習しない。TD 誤差は，

$$\begin{aligned}\hat{r}_t &= Q_{train,a_t,t} - Q_{a_t}(\boldsymbol{s}_t) \\ &= r_{t+1} + \gamma \max_a Q_a(\boldsymbol{s}_{t+1}) - Q_{a_t}(\boldsymbol{s}_t). \cdots\cdots\cdots (2)\end{aligned}$$

のように定義される。Actor の出力に対する教師信号は

$$\boldsymbol{m}_{train,t} = \boldsymbol{m}(\boldsymbol{s}_t) + \hat{r}_t \boldsymbol{rnd}_t \cdots\cdots\cdots\cdots\cdots (3)$$

とする。ここで，$\boldsymbol{m}(\boldsymbol{s}_t)$ は $\boldsymbol{s}_t$ が RNN の入力の際の Actor の出力ベクトルであり，$\boldsymbol{rnd}_t$ は前述の探索ベクトルである。そして，$Q_{train,a_t,t}$ と $\boldsymbol{m}_{train,t}$ を教師信号として用い，$\boldsymbol{s}_t$ が入力の場合の RNN を BPTT(Back Propagation Through Time)[20] で1回だけ学習する。ここでは，値域が -0.5 から 0.5 のシグモイド関数を各中間層および出力層ニューロンの出力関数として用いたため，値域を調整するために，ニューロンの出力の値域 $[-0.5, 0.5]$ と Q 値の値域 $[-0.2, 0.8]$ との間はシフトして用いる。Actor の出力は変換せず，そのまま用いる。この学習で重要なことは，学習方法はとても単純で汎用的であり，与えたタスクに対する特別な学習は行っていないということである。

3. シミュレーション

〈3・1〉 不可視物体捕獲タスク

図1を用いてタスクの説明を行う。8.0×3.0 の大きさのフィールドがあり，その左端から物体が移動を開始する。初期の y 座標 $p_{y,0}$，x 方向の初速度 $v_{x,0}$，移動方向の x 軸からの角度 θ_0 は，図のように，試行ごとにランダムに決定される。ここで，速度は1ステップの移動量とする。物体はまっすぐ進み，$y = 0.0$ または $y = 3.0$ に着くと壁に衝突し，x 方向と y 方向の速度 v_x，v_y は，それぞれ 0.9 倍，-0.8 倍に変化する。エージェントは最初は $(7.0, 1.5)$ に位置する。前述のように，出力は Q 値用と Actor 用の2つに分けられる。ここでは,「移動」と「捕獲」の2つの行動に対応した2つの Q 値の出力があり，ボルツマン選択[21] によって Q 値に応じた確率で一つの行動が選択される。また，2つの Actor の出力があり,「移動」を選んだ場合には，これに探索のための一様乱数をそれぞれ加えた連続値 $v_{agent,x}$，$v_{agent,y}$ だけ x 方向および y 方向に移動する。1回の移動は x 方向，y 方向それぞれ，物体の x 方向の移動速度 v_x の最大値の半分である ± 0.4 に制限される。「捕獲」の行動を選択した場合は移動せず，次のステップで物体を捕獲する。エージェントと物体間の距離 $dist$ が $dist < 1.0$ かつ，物体の x 座標 p_x が $p_x > 6.0$ であれば，エージェントは $0.7 * (1.0 - dist^2)$ の報酬を得て，その試行

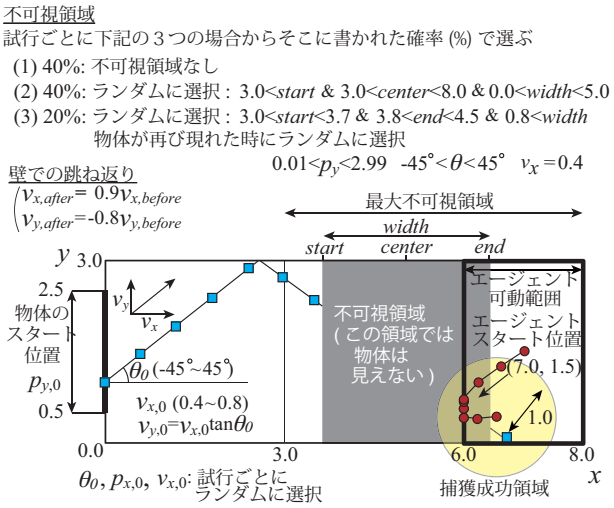


図 1　不可視物体捕獲タスク

は終了する。ただし，物体が $p_x = 6.0$ の前と後での報酬の値の急激な変化を避けるため，$6.0 < p_x \leq 6.1$ で物体を捕獲した場合は，報酬の値を $(p_x - 6.0) * 10$ 倍に割り引いて滑らかに変化させた。もし，$p_x < 6.0$ または $dist > 1.0$ でエージェントが「捕獲」を選択した場合や $p_x > 8.0$ になるまで「捕獲」を選ばなかった場合には -0.1 の小さな罰を与えた。学習時は，学習の加速のために物体が $p_x > 8.0$ に至るまで試行は終了しなかったが，学習後のテストフェーズでは，そこで試行を終了した。

このタスクを予測が必要なタスクとするために不可視領域を設け，物体がしばしばフィールド内で見えなくなるようにした。図 1 の上部に書いたように，確率 40%で不可視領域なし，40%は x 座標が 3.0 より大きい場所に現れ，非常に狭い範囲の場合から，最後まで物体が見えない場合まで含めてランダムに設定した。残りの 20%でも不可視領域が現れるが，x 座標が 4.5 より前で終了し，再度出現した物体は，見えなくなる前の運動とは関係なく，v_x を最小値の 0.4 に，出現位置，移動方向をランダムに決定した。したがって，物体が見えない場合は，物体の運動の予測に基づいて捕獲の位置とタイミングを決めなければならず，また物体が再び現れて，前の運動と矛盾している時には，即座に戦略を切り替えて，将来の予測をし直して，捕獲の位置とタイミングを考えなくてはならない。しかしながら，エージェントは不可視領域の情報を予め与えられないため，物体がいつ消えて，いつ現れるかを知ることはできない。

図 2 に，使用する RNN の入力，出力，構造を示す。物体用とエージェント自身用の２つの固定された視覚センサを置いた。センサセルは２次元平面上に x 方向，y 方向それぞれ 0.2 の間隔で配置されるので，物体の位置用として $41 \times 16 = 656$ の入力が，エージェントの位置用として $11 \times 16 = 176$ の入力が RNN に与えられる。各セルは局所的な小さな受容野を持ち，その出力は物体またはエージェントの位置を入力とするガウシアンとして計算される。

表 1　学習に使用したパラーメータ。‘→’ は，学習とともに直線的に減少させたことを表す。

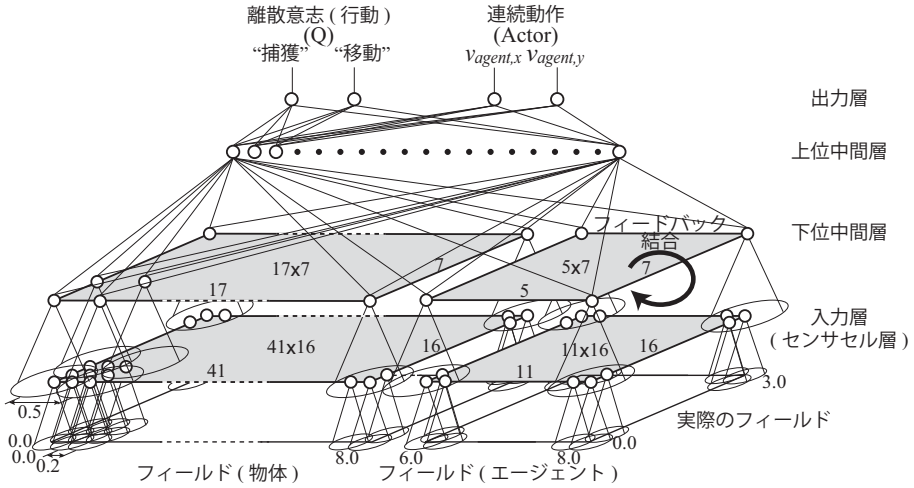
学習係数 (フィードバック)	0.02→0.01	ボルツマン選択の温度	0.05→0.025
(その他)	0.1→0.05	Actor のための探索 (一様乱数) の大きさ	±0.5→±0.15
割引率	0.96		

下位の中間層ニューロンも２次元平面上に配置され，各ニューロンは最初局所的な領域に反応するようにした。つまり，図 2 のように，入力信号との初期重み値は当該中間層ニューロンと入力ニューロンとの距離を入力としたガウシアンとして計算して与えた。これらのニューロンは，Elman ネットのように，この層のすべてのニューロンからフィードバック結合を受ける。誤差信号が効果的かつ発散しないで伝播するように，セルフフィードバック結合の初期重み値は 4.0 とし，その他のフィードバック結合については 0.0 とした。下位と上位の中間層ニューロン間の結合の初期重み値はすべて 0.0 とした。各ニューロンは，入力がすべて 0.0 であると出力も 0.0 となるので，学習前は，どのような入力に対しても RNN の出力は 0.0 となる。上位中間層ニューロンと出力ニューロン間の初期重み値は ±0.1 の範囲の乱数とし，上位中間層ニューロンには，学習を通して，物体とエージェントに関する情報を統合することを期待する。

この学習で用いたパラメータを表 1 に示す。

〈3･2〉 結　　果　図 3 に学習曲線を示す。(a) と (b) はそれぞれ，学習の進行に伴う成功確率と平均獲得報酬の変化を示す。探索要素は，離散行動，連続動作ともに試行回数とともに変化させているので，その影響を参照するために，学習後の重み値を用いたときの探索要素のみによる変化もそれぞれのグラフに一緒にプロットした。(b) では 2 種類の学習曲線を示す。一つは成功した試行だけの平均報酬，もう一つはすべての試行の平均報酬であり，いずれも 1,000 試行分の平均である。最初，成功率はほとんど 0 であり，ほとんどの場合でエージェントは，物体が $p_x > 6.0$ の範囲に入る前に「捕獲」の行動を選択したが，成功率は徐々に上がり，最後はほぼ 100%になった。成功試行の平均報酬は最初 0.4 程度である。エージェントから 1.0 の距離の捕獲成功範囲内で捕獲場所がランダムであったとすると，その期待報酬は最大報酬の半分の 0.35 となる。学習が進むにしたがって，平均報酬は徐々に大きくなった。学習後，ランダムな初期状態から 10,000 試行行ったところ，失敗したのは 4 試行だけであった。そのうち 1 回は，物体が $x > 6.0$ になる前に「捕獲」行動を選んでしまい，その他の 3 回の場合は，物体との距離が 1.0 以上あった。平均報酬は 0.656 であり，物体との平均距離は 0.21 であった。

初期重み値や各試行での物体の動き，不可視領域，およびエージェントの探索成分を決めるために使われる乱数系列を変えて 10 回学習を行った。学習後，10,000 試行中の平均失敗回数は 9.0 (0.09% SD 5.2) で，平均報酬は 0.652



層	ニューロン数	初期重み値
出力層	4	乱数 -0.1 ～ 0.1
上位中間層	30	0.0
下位中間層	154	(入力層から) 局所結合 (セルフ フィードバック) 4.0 (その他の フィードバック) 0.0
入力層	832	—

図 2　リカレントニューラルネットを用いた Actor-Q 学習。右の表は各層のニューロン数，初期重み値。

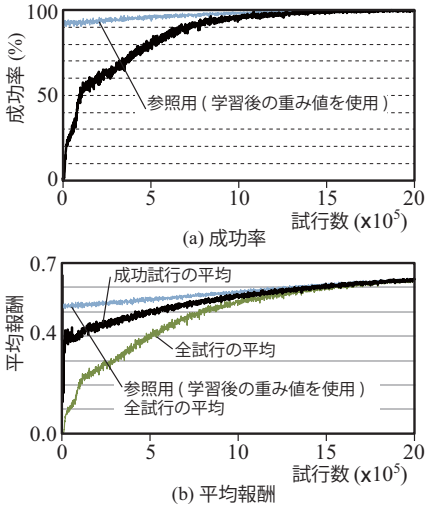


図 3　学習曲線（(a) 成功率の変化, (b) 平均報酬の変化）

(SD 0.003) であった。図 2 のような RNN の入力－下位中間層間の局所初期結合を導入しなかった場合は平均失敗回数は 57.6 (SD 20.7) で平均報酬は 0.630 (SD 0.007) であった。さらに，初期のセルフフィードバック結合を 4.0 の代わりに 0.0 にすると，学習は非常に遅くなり，失敗回数は 1,687 回と増加した。また，BPTT で，フィードバック結合をさかのぼる誤差信号の伝播を行わなかった場合は，失敗回数は 2,312 回に上った。後の 2 つの場合，この論文に結果を載せた場合と同じ初期重み値を用いて 1 回だけシミュレーションを行った。これらの結果から，下位中間層と入力層の間の局所初期重み値および BPTT による過去にさかのぼっての学習が有効に働いていると考えられる。

図 4(a) に，不可視領域最大で物体が再度現れない場合の 4 つの学習後のサンプル軌道を示した。ケース 1 では，物体の速度の x 方向成分 $v_{x,0}$ が最小の 0.4 で，エージェントが到達しなければならない場所が $y = 0$ と下の壁に近い。ケース 2 では，物体の初期位置 $p_{y,0}$ と $v_{x,0}$ は前と同じであるが，物体の移動方向 θ が異なっており，スタート直後はケース 1 より y 座標の増加が小さいが，最終的にエージェントは上の壁である $y = 3$ の方へ行かなければならない。ケース 3 では，$p_{y,0}$ と θ_0 はケース 1 と同じであるが，$v_{x,0}$ が最大の 0.8 となっており，物体を捕獲するタイミングが大きく異なる。ケース 4 では，物体は壁と平行に最大速度で進み，消える前の物体の位置はケース 3 と近いが，捕獲位置は全く異なる。いずれの場合も，エージェントは見えない物体に近づき，捕獲に成功していることがわかる。軌道を観察すると，エージェントは最初に可動範囲の左端の $x = 6.0$ まで行き，物体が近づくとエージェントが物体より先に右に移動を始めていることがわかる。他のシミュレーションでも，同様な傾向が見られた。これは，エージェントが $x = 8.0$ 周辺で物体を捕獲するよりは $x = 6.0$ 周辺で捕獲したほうが短いステップ数で捕獲できること，さらに，ボルツマン選択により，エージェントが「捕獲」行動を選ばなかった場合でも，物体と同じ方向に移動することで，次のステップで物体を捕獲できるように学習したと考えられる。

図 4(b) は RNN の出力である Q 値の 1 試行の間での変化を，物体の速度だけが違うケース 1 と 3 の場合について示した。いずれの場合も，早すぎるタイミングで「捕獲」を選んだときの罰により,「捕獲」の Q 値は最初は非常に小さい値になっている。「移動」の Q 値は徐々に増加しているが，物体が近づくと，まるで見えているかのように「捕獲」の Q 値が急激に上昇し，最後は「移動」の Q 値を追い越して捕獲に至っている。$p_x > 8.0$ での罰により,「移動」の Q 値は最後は減少している。各ステップでの大きい方の Q 値は徐々に増加しており，その変化は割引率 0.96 から求められる理想値に近い。また，いずれの場合も「移動」の初期 Q 値は 0.4 付近であるが，1 ステップ後には物体の x 方向速度 $v_{x,0}$ に応じて異なる値になっており，エージェントはすぐに捕獲タイミングの違いを予測していることがわかる。

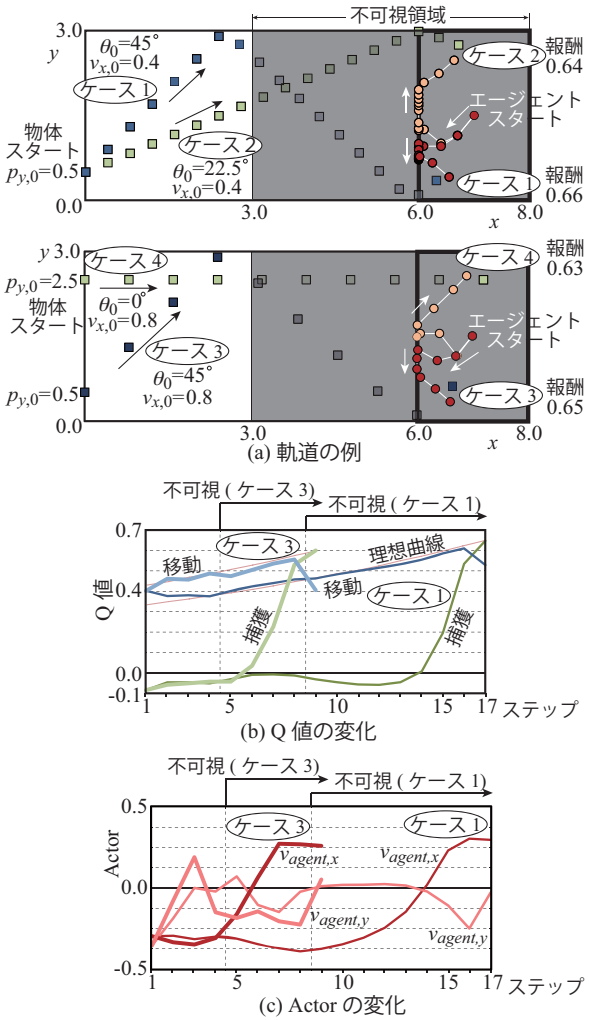


図 4　学習後の物体とエージェントの軌道例 (a) と (a) のうちの速度だけ違う 2 つの場合の Q 値と Actor の変化 (b,c)

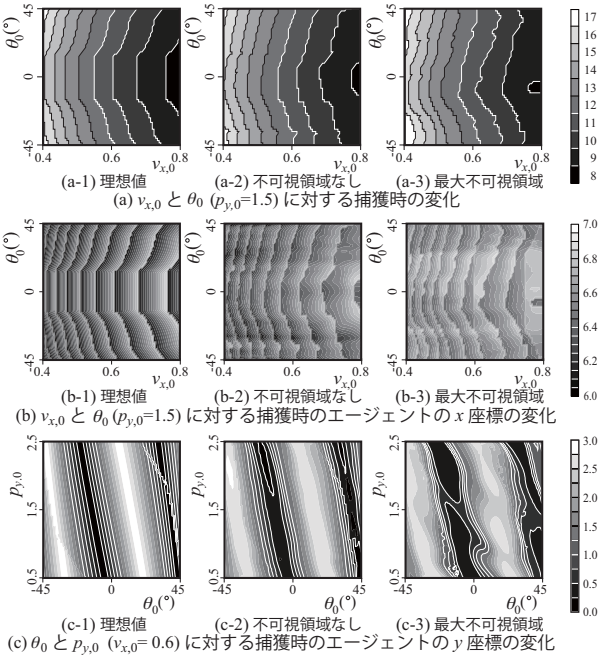


図 5　理想の場合を含めた 3 つの場合についての学習後の x 方向速度 $v_{x,0}$ と移動方向 θ_0 に対する (a) 捕獲タイミングと (b) そのときのエージェントの x 座標および θ_0 と y 座標 $p_{y,0}$ に対する (c) 捕獲時のエージェントの y 座標。各図は学習後の 81×61 回のテスト試行によって作成した。

図 4(c) は Actor の出力を表している。(a) の軌道からもわかるように，エージェントの速度の x 方向成分 $v_{agent,x}$ は最初は負で,「捕獲」の Q 値が上昇する前に上昇し始めている。ケース 1 の 17 ステップ目，および，ケース 3 の 9 ステップ目では,「捕獲」の Q 値が選ばれるので実際はエージェントは動かないが，探索要素によって「移動」行動が選択されると，この Actor 出力が使われる。

図 5 は，初期の物体の速度の x 方向成分 $v_{x,0}$ と移動方向 θ_0 に対する (a) 捕獲タイミング，(b) 捕獲時のエージェントの x 座標を 3 つの場合について示した。一番左の最初のグラフは理想値を示している。2 つ目と 3 つ目のグラフでは，不可視領域がない場合と不可視領域が最大で物体が再度現れることがない場合の実際の値を示している。(c) では，捕獲時のエージェントの y 座標を，物体の初期の移動方向 θ_0 と初期の y 座標 $p_{y,0}$ を変化させてプロットした。(a) では，最も長い 17 ステップかかる試行が最も短い場合の 8 ステップの 2 倍以上あるにもかかわらず，(a-2) と (a-3) の場合は (a-1) の場合にとても良く似ている。特に，移動方向 θ_0 の絶対値が大きくなるにしたがって，(a-2) の場合も (a-3) の場合も，物体の壁との衝突を考慮して捕獲タイミングは遅くしていることがわかる。(a-3) では，ステップ数は全体的に理想値より少し大きくなっている。それは，不可視領域のために物体の位置をあまり正確に把握できないため，$p_x < 6.0$ の罰をもらう領域や $6.0 \leq p_x \leq 6.1$ の報酬が小さくなっている領域を避けようとしたためと考えられる。

(b) からは，捕獲タイミングが同じであっても，捕獲時の x 座標が物体の x 方向の初速度 $v_{x,0}$ によって変化していることがわかる。(b-3) で全体的に x 座標が大きかったことは，(a-3) で捕獲が遅くなることと同様な理由であると推測される。(b-3) では，$v_x = 0.5, 0.6, 0.75$ で垂直な線が見えるが，これは，$x = 3.0$ の辺りで v_x の小さな変化によって物体が不可視領域に入ったり入らなかったりする違いによって，捕獲の x 座標が異なることによるものである。

(c) では，特に (c-3) は (c-1) と異なって見えるものの，$p_{y,0}$ と θ_0 の変化による y 座標の傾向は (c-3) においても反映されている。また，物体を捕獲するために壁の近く，つまり，$y = 0$ や $y = 3$ の近くに行かないことがわかる。これは，ときに予測結果と物体が矛盾した動きをすることがあること，さらには，物体が $y < 0.0$ や $y > 3.0$ に行くことがないことが理由と考えられる。

最後に，図 6 は，不可視領域が $x = 3$ から $x = 4.5$ までで，物体が再び現れた時の動きが消える前と矛盾した場合

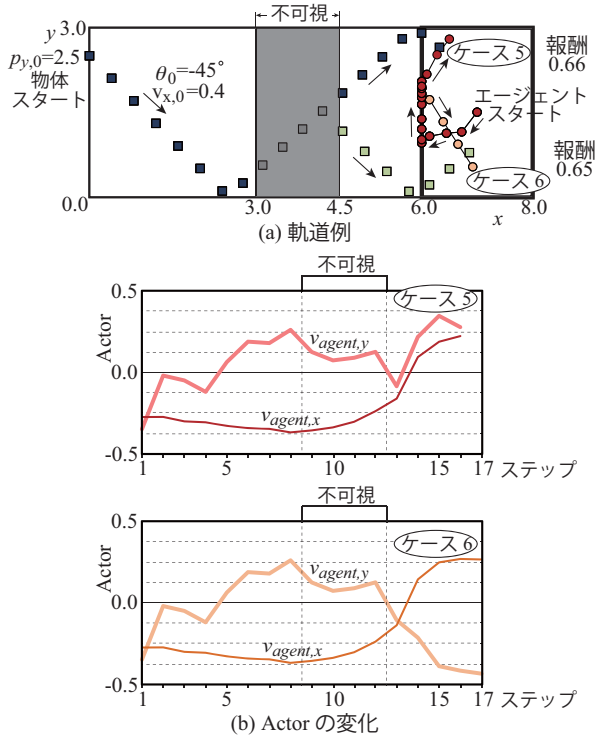


図 6　物体再出現時の物体の動きの変化によるエージェントの軌道と Actor の変化

にエージェントの行動がどう変化したかの例を示す。ケース 5 では，物体の動きは見えなくなる前と矛盾なく，エージェントはフィールドの上の方で物体を捕らえる必要がある。一方，ケース 6 では，物体の y 方向の速度は再び現れた時は反転され，エージェントは最初の予測とは逆のフィールドの下の方に行かなくてはならない。エージェントは最初 $x = 6.0$ に近づき，その後，少し上に行ったが，上の壁にはあまり近づかなかった。物体が再び現れた後，２つの場合の軌道はすぐに大きく分かれ，いずれの場合も物体を捕獲することができた。(b) を見ても，物体が現れた後，エージェントの y 方向動作を表す Actor である $v_{agent,y}$ がケース 5 では正に，ケース 6 では負になっていることがわかる。ケース 6 では，捕獲タイミングがケース 5 の場合より遅い。エージェントは，16 ステップ目で到達することができないため，正の $v_{agent,x}$ で移動し，17 ステップ目で捕獲することを選んだと考えられる。また，エージェントは試行の初期に壁のあまり近くに行かないが，これは，予測と矛盾した物体の動きがありうるからであると推測される。

4. 結　論

本稿では,「不可視物体捕獲」タスクにおいて，予測を必要とする離散意志決定と連続値動作の両方を,「予測が必要である」とか「どんな情報を予測すべきか」を与えることなく，リカレントニューラルネットを用いた強化学習を行うだけで学習できることを確認した。また，このタスクでは，物体は時々見えない間にランダムに移動方向を変えるため，戦略の切り替えが求められるが，特別な機構や手法を追加することなく，学習を通して柔軟な行動を自律的に獲得することができた。報酬と罰からの学習だけで，事前知識なくエージェントやロボットがこのような柔軟な行動を獲得する創発の特性は，一般的な従来の並列処理システムでは実現できないものであり,「フレーム問題」,”What” 問題を根本的に解決する鍵となると著者らは信じている。

参考文献

(1) Dennett, D. Cognitive Wheels : The Frame Problem of AI. *The Philosophy of Artificial Intelligence*, Margaret A. Boden, pp. 147-170, Oxford University Press, 1984.

(2) R. A. Brooks, Intelligence without Representation. *Artificial Intelligence*, **47**, pp.139-159, 1991.

(3) K. Shibata, Emergence of Intelligence through Reinforcement Learning with a Neural Network. *Advances in Reinforcement Learning*, InTech, pp.99-120, 2011.

(4) L.-J. Lin, & T. M. Mitchell, Reinforcement learning with hidden states, *From Animals to Animats 2*, pp. 271-280, MIT Press, 1993.

(5) A. Onat, et al., Q-Learning with Recurrent Neural Networks ..., *Proc. of ICONIP 98*, pp. 837-840, 1998.

(6) H. Arie, et al., Reinforcement learning of a continuous .., *Advanced Robotics*, **21** (10), pp. 1215-1229, 2007.

(7) A. Onat, et al., Reinforcement learning of dynamic behavior ..., *Artificial Life Robotics*, **1**, pp. 117-121, 1997.

(8) B. Bakker, et al., A Robot that Reinforcement-Learns to Identify ..., *Proc. of IROS 2003*, pp. 430-435, 2003.

(9) K. Shibata, Discretization of Series of Communication ..., *Adaptive and Natural Computing Algorithms*, pp. 486-489, 2005.

(10) H. Utsunomiya & K. Shibata, Contextual Behavior and Internal ..., *Advances in Neuro-Information Processing*, LNCS, **5506**, pp. 755-762, 2009

(11) K. Goto & K. Shibata, Acquisition of Deterministic Exploration ..., *Proc. of SICE Annual Conf. 2010*, 2010.

(12) K. Shibata, et al., Discovery of Pattern Meaning from Delayed ..., *Proc. of IJCNN. 2011*, pp. 1445-1452,2011.

(13) J. Schmidhuber, Reinforcement learning in Markovian ..., *Advances in NIPS 3*, pp. 500-506, 1991.

(14) J. Tani, Learning to generate articulated behavior ..., *Neural Networks*, **16**(1), pp. 11-23, 2003.

(15) K. Goto, & K. Shibata, Emergence of prediction ..., *Journal of Robotics*, **2010**, Article ID 437654, 2010.

(16) K. Shibata & K. Goto, Emergence of Flexible Prediction-based Discrete Decision..., *Proc. ICDL-Epirob 2013*, 2013 (to appear)

(17) C. J. C. H. Watkins, Learning from Delayed Rewards, *PhD thesis*, Cambridge University, England, 1989.

(18) K. Shibata, et al. Active Perception and ..., *Systems and Computers in Japan*, **33**(14), pp. 12-22, 2002.

(19) A. G. Barto, et al., Neuronlike adaptive elements that ..., *IEEE Trans. on SMC*, **13**(5), pp. 834-846, 1983.

(20) D. E. Rumelhart, et al., Learning Internal Representation by Error Propagation, *Parallel Distributed Processing*, MIT Press, **1**, pp. 318-364, 1986.

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