日本機械学会論文集(C編)
70巻694号(2004-6)

論文 No. 03-1210

自己組織化マップを用いた教示による強化学習の高速化手法の提案*

舘山武史*1, 川田誠一*1, 小口俊樹*1

A Teaching Method by Using Self-Organizing Map for Reinforcement Learning

Takeshi TATEYAMA*2, Seiichi KAWATA and Toshiki OGUCHI

*2 Department of Mechanical Engineering, Graduate School of Engineering, Tokyo Metropolitan University, 1-1 Minami-Ohsawa, Hachioji-shi, Tokyo, 192-0397 Japan

A new pre-teaching method for reinforcement learning using Self-Organizing Map (SOM) is described. The purpose of our study is to increase the learning rate using small number of teaching data generated by a human expert. In our method, the SOM is used to generate initial teaching data for the reinforcement learning agent from a few teaching data. The reinforcement learning function of the agent is initialized by using the teaching data generated by the SOM so as to increase the probability of selecting the optimal actions estimated by the SOM. Because the agent can get high rewards from the start of reinforcement learning, it is expected to increase the learning rate. The results of two computer simulations, mobile robot navigation and pursuit game, showed that the learning rate increased although the human expert had showed only a few teaching data.

Key Words: Reinforcement Learning, Teaching, Self-Organizing Map, Q-Learning, Actor-Critic Method

1. はじめに

強化学習法 (Reinforcement Learning) は, 学習エージェントが未知の環境内で試行錯誤を繰り返すことにより環境に適応した制御法を学習していく手法であるが, 状態数が多い環境では膨大な学習時間が必要となることが問題である. 一般的に, 強化学習法は環境やタスクに関する知識を全く持たない状態から試行錯誤を行うため, 特に学習の初期段階では報酬の獲得に膨大な試行が必要となる. 特に実システムで学習を行う場合は, 行動の実行に多大な時間やコストがかかることも考えられるため, 強化学習法の実用化のためには, 学習の高速化は極めて重要な課題であるといえる.

強化学習法の収束を早める手段の一つとしては, 人間による教示を導入した研究例があり, その有効性が示されている[1].Lin は初期状態からゴール状態までの"lessons"と呼ばれる一連のシーケンスを人間が作成して教示する手法を提案し, 移動ロボットの迷路タスクのシミュレーションによってその有効性を示している[2]. また, Clouse らは教示が必要であると教師が判断したときに随時オンラインで教示信号を与える手法を提案しており[3], 倒立振子の学習の収束を早めることに成功している. このように, 本来教師無しの学習法である強化学習に教示を導入することに関しては意見が分かれるところだが, Kaelbling らは既存のほとんどの強化学習法では大規模な問題には適用が困難であることを指摘し, 学習の補助になるような偏り (模範, 問題分割など) を与えるべきであるとしている[4].

本研究では, 強化学習の収束を早めるための学習補助としては, 教示の導入が非常に有効な手段であると考え, 過去の研究例の欠点を補うことにより, より実用的な教示の導入法を開発することを目的としている.Lin の手法は, タスクが達成できる適当なシーケンスを時間オーダーで人間が生成しなくてはならないため, 問題が複雑な場合は教師 (人間) への負担が大きくなってしまう. 一方, Clouse のオンライン教示法は, 問題によっては教示を行うタイミング等に教師の熟練が必要となることも考えられ, この手法もまた教師への負担が大きくなることが考えられる.

本研究では, 人間が与える教示を最小限の代表値に限定し, 負担を軽減することを提案する. まず代表点の教示を自己組織化マップ (SOM)[5]の入力とし, SOM に学習させる. そして, その結果得られたより広い状態空間への教示結果を強化学習の初期値とすることで, 総合的な学習の効率を高めることを目的とする.SOM のニューロンの重みベクトルを状態と行動のベクトルに対応させる先行研究としては, Sehad らの研究[6]がある. 彼らは重みベクトルに状態, 行動と Q 値を対応さ

* 原稿受付 2003年10月20日.
*1 正員, 東京都立大学大学院工学研究科(〒192-0397 八王子市南大沢1-1).
E-mail: tateyama@control.prec.metro-u.ac.jp

せ, 試行錯誤によってそれらの値を更新し, 最適な状態と行動の組を求めているが, 本研究のように教示を導入することは考慮に入れていない. 本研究では, Q値はSOMでは扱わず, 状態と行動のベクトルを教示データとしてSOMへ入力して学習させ, 強化学習を行なう前の最適行動予測器として利用することを提案する. この最適行動予測器は, 強化学習時の初期値の設定に利用されるが, 本提案手法はさまざまな強化学習法と組み合わせることが可能である.

本論文では, まず2章で強化学習法の基礎理論を述べ, 3章で本提案手法の詳細について述べる. そして4章では移動ロボットの最短経路探索タスク, 5章でマルチエージェント追跡問題[7]のシミュレーションへ本手法を適用し, その有効性を確認する. 最後に, 6章において結論と今後の課題について述べる.

2. 強化学習法

2·1 Q-learning ここでは, 代表的な強化学習アルゴリズムの一つであるQ-learning[8]について解説する. 環境内における状態集合をS, エージェントが実行可能な行動の集合をAとする. 時刻tで観測された状態$s_t \in S$においてエージェントが行動$a_t \in A$を実行し, 時刻$t+1$で状態s_{t+1}に推移し, 報酬r_{t+1}を得たとすると, 状態s_tの行動価値関数$Q(s_t, a_t)$は, 次式を用いて更新される.

$$Q(s_t,a_t) \leftarrow (1-\alpha)Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1},a)] \quad (1)$$

ここで, $\alpha(0<\alpha\leq 1)$は学習率, $\gamma(0\leq\gamma<1)$は割引率である. 行動価値関数とは状態と行動の組に対する評価を見積もる関数であり, Q-learningではその評価値をQ値と呼ぶ. エージェントは学習したQ値の大きさに関係した確率で行動を決定するが, その行動選択法としては, ε-greedy法や, Boltzmann分布を用いた手法などがある[8].

2·2 actor-critic法 actor-critic法[8][9]は, 状態の価値を評価するcriticと, 状態に応じて行動を選択するactorの2要素から構成される, 強化学習アルゴリズムの一つであり, 連続値の行動を扱う場合に有効な手法である. エージェントは時刻tにおいて状態s_tを観測し, それに応じてactorが確率的政策$\pi_t(s_t,a)$に従って行動a_tを実行する. エージェントは次の時刻$t+1$で状態s_{t+1}に推移し, 報酬r_{t+1}を受け取るが, このときcriticは次式を用いてTD誤差 (TD-error)δ_tを計算し, 状態s_tの評価をする.

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \quad (2)$$

TD誤差は, 推移先の価値の推定値$V(s_{t+1})$を用いて表される$V(s)$の目標値と, 状態s_tの真の価値である$V^\pi(s_t)$の推定値$V(s_t)$との差を表している. もしTD誤差が正ならば, 実行した行動a_tは比較的好ましい行動と考えられるので, この行動の選択確率を上げる. 逆にTD誤差が負ならば, 行動a_tは比較的好ましくないと考えられるので, この行動の選択確率を下げる. 状態s_tで行動a_tを選択する確率$p(s_t,a_t)$の更新式の例を以下に示す.

$$p(s_t,a_t) \leftarrow p(s_t,a_t) + \alpha_a \delta_t \quad (3)$$

ここで, $\alpha_a(0<\alpha_a\leq 1)$はactorの学習率である. そして, criticが評価した状態s_tの価値$V(s_t)$は, 次式を用いて更新される.

$$V(s_t) \leftarrow V(s_t) + \alpha_c \delta_t \quad (4)$$

ここで, $\alpha_c(0<\alpha_c\leq 1)$はcriticの学習率である.

3. 自己組織化マップ (SOM) を用いた強化学習の効率的な教示法

3·1 提案手法の概要 先に述べたように, 強化学習法に教示を導入する手法はこれまでにいくつか提案されているが, 本研究では以下の項目に重点を置いている.

- 教師 (人間) が作成する教示データ (状態-行動の組) の数をできるだけ少ないものとし, 教師の負担を軽くする. しかも, 少ない教示データで最大限の教示の効果が得られるようにする.
- 教示データを学習エージェントに与えるのは強化学習を行う前の初期設定の時のみとし, Clouseの手法[3]のように, 学習中に教師がエージェントの挙動を常に監視しなければならないようなことは避ける.
- 教師が与えた教示の内容は学習時間の経過とともに効果が薄れていくようにし (学習中には教示を行わないことにより, 教示内容が徐々に修正されていくことを意味する), 学習の初期段階の無駄な試行錯誤の削減に重点を置く. これにより, 教示内容が誤っていた場合でもエージェントは学習により最適政策を取得することが可能となる.

以上のような目標を達成するために, 我々は自己組織化マップ (Self-Organizing Maps, SOM) を用いた効率的な教示法を提案する. 本提案手法では, まず教師が作成した代表状態の教示データをSOMの入力として, SOMに学習させる. そして, その学習によって得られたより広い状態空間への教示結果を強化学習の初期設

定 (Q-learning の場合はQ値の初期値設定, actor-critic 法の場合は actor の行動選択確率の初期設定) に反映させることにより, 強化学習の学習効率を向上させることがねらいである. 本研究では, SOM は与えられた教師データの関数近似を行う要素として用いている. この他に線形近似や一般的なニューラルネットワークを用いることも考えられるが, 複雑な関数近似や多次元, 多数の教示データを用いることを視野に入れ, 本研究では SOM を適用することにした.

強化学習法とは本来, 未知環境内において人間による制御設計が困難であるような複雑なタスクを学習する事を目標としているため, 教示の導入の有効性に関しては意見が分かれるところもあるが, Kaelbling らは文献[4]において, 既存の強化学習法を大規模な問題へ適用することの困難さを指摘し, 模範, 問題分割など, 人間が学習の補助になるような偏りを与えるべきであるとしている. また, 前述の Lin や Clouse の実験の例のように, タスクの部分教示が可能で, かつその後の強化学習が必要となる状況も十分考えられる. 本研究は, 大規模, 複雑なタスクの学習には教示の導入が非常に有効であるという考えに基づき, その有効性, 効率性をさらに高めようというものである.

3·2 提案手法の詳細 図1に本提案手法のシステム構成図を示す. 以下に, 各項目の詳細を述べる.

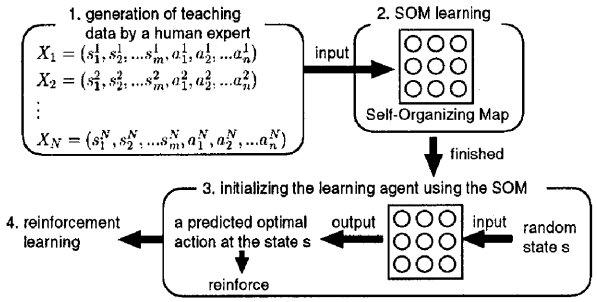

Fig. 1 The outline of the proposed method

3·2·1 教示データの作成 まず, 教師が比較的良いと思われる状態-行動の組を予想する. ここで, 学習エージェントが観測する状態ベクトルが m 次元のベクトルであり, 実行する行動が n 次元のベクトルで表されるとすると, 教師が作成する教示データ X は, 以下のような $m+n$ 次元のベクトルで表される.

$$X = (s_1, s_2, ..., s_m, a_1, a_2, ..., a_n) \tag{5}$$

SOM への入力データとなるベクトル X は, 状態 $s = (s_1, s_2, ..., s_m)$ において行動 $a = (a_1, a_2, ..., a_n)$ を実行することが良いと予想されることを表している. 教示データは, 任意の数 N だけ作成する.

3·2·2 SOM による学習 SOM は T.Kohonen が提案した, 汎化能力が優れたニューラルネットワークであり, 主に画像解析, 音声解析等のデータ解析に応用されている. 本研究では, この SOM の優れた学習能力を利用し, 教師が作成した限られた数の状態と行動の組で表される教示データを学習させ, より広い状態空間で教示の効果が得られるようなシステムを構築する.SOM のニューロンの重みベクトルを状態と行動のベクトルに対応させる先行研究としては, [6]がある. Sehad は重みベクトルに状態, 行動と Q 値を対応させ, 試行錯誤によってそれらの値を更新し, 最適な状態と行動の組を求めている. Sehad のシステムは強化学習を行なうシステムであるが, 本提案手法は強化学習前に最適な状態と行動の組を予想する最適行動予測器として SOM を用いるため, ニューロンの重みベクトルは状態, 行動ベクトルのみ表現するものとする.

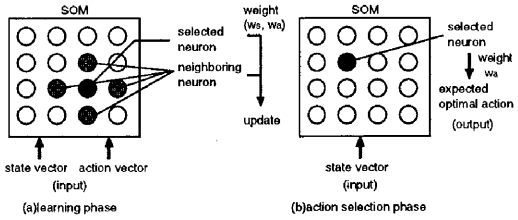

Fig. 2 SOM's learning phase(a) and action selection phase(b)

図2は SOM の学習段階 (a) と行動選択段階 (b) の概略図である. 学習段階では, SOM への入力は教示データのベクトル s と a であり, N 個のそれぞれの教示データについて繰り返し学習を行う. 以下に, 学習手順を示す. まず, 作成した教示データ (s, a) を入力ベクトルとして SOM に入力し, 次式を用いて j 番目のニューロンと入力ベクトル X とのユークリッド距離 d_j を求める.

$$d_j = \|X - W_j\| = \sqrt{\sum_{k=1}^{m}(s_k - w_{jk})^2 + \sum_{l=1}^{n}(a_l - w_{jm+l})^2} \tag{6}$$

ここで W_j はニューロン j の重みベクトルであり, 以下に示すような $m+n$ 次元の要素で構成される.

$$W_j = (w_{j1}, w_{j2}, ..., w_{jm}, w_{jm+1}, ..., w_{jm+n}) \tag{7}$$

こうして計算された距離 d_j が最小であるニューロンが発火ニューロン (selected neuron)c として選択される. そして, 発火ニューロンとその付近のニューロン (neighboring neurons) の重みベクトルを, 次式を用いて入力ベクトルに近づけるように更新する.

$$w_{jk} \leftarrow w_{jk} + \alpha(t)(s_k - w_{jk}) \tag{8}$$

$$w_{jm+l} \leftarrow w_{jm+l} + \alpha(t)(a_l - w_{jm+l}) \quad (9)$$
$$(k = 1,2,...,m,\ \ l = 1,2,...,n)$$

ただし, $\alpha(t)$ は SOM の学習率 $(0 < \alpha(t) \leq 1)$ であり, 一般には時間の経過と共に減少させていく方法をとる.

次に, 学習が終了した SOM を用いて任意の状態の最適行動を予想する. このとき, 状態 s における予想最適行動を SOM によって算出するには, 2(b) に示すように, SOM には状態ベクトル s のみを入力する. そして, 次式を用いて再び各ニューロンと入力ベクトルとの距離 d'_j を求める.

$$d'_j = \|s - W_{js}\| = \sqrt{\sum_{k=1}^{m}(s_k - w_{jk})^2} \quad (10)$$

ここで, W_{js} はニューロン j の重みベクトル W_j の状態に対応するベクトルを表している. こうして, 任意の状態 s の予想最適行動は, 距離 d'_j が最小であるニューロン c' の重みベクトル

$$W_{c'a} = (w_{c'm+1}, w_{c'm+2}, \cdots, w_{c'm+n}) \quad (11)$$

を読みとることで求められる.

3·2·3 教示内容を強化学習の初期設定に反映 次に, SOM の学習によって得られた教示データを用いて, 強化学習を行うエージェントの学習初期段階における行動選択に偏りをもたせる. つまり, 良い行動であると予想された行動が選択されやすいようにエージェントの初期設定を行う. 例えば, 強化学習アルゴリズムに Q-learning を用いる場合は, タスク実行に有効であると予想される状態-行動の組の Q 値を他の組の Q 値よりも高めの初期値を設定する. ここで, 任意の状態 s の予想最適行動ベクトルは, 前述のように状態ベクトル s を入力とする SOM の発火ニューロン c' の重みベクトル $W_{c'a}$ を算出することにより求められる.

3·2·4 強化学習 エージェントの初期設定が完了した後は, 通常のアルゴリズムに従って強化学習を行う. 予め有効であると予想される行動が選択されやすくなっているため, 報酬獲得につながる行動を発見しやすく, 結果として学習の収束が早まることが期待できる. また, 仮に教示データが誤っていた場合でも, エージェントは試行錯誤によって自分自身で政策を改善していくことが可能である.

4. 移動ロボットの迷路走行タスクへの適用

4·1 問題設定 ここでは, 移動ロボットの迷路走行タスクへの本手法の適用例を示す. 環境は図 3 に示すような 90[m] × 60[m] の外壁に囲まれた部屋であり, 3 つの障害物が配置されている. 学習エージェントとなる移動ロボットは 1[sec] ごとに現在の座標 (x[m], y[m]) を状態として観測し (部屋の南西端を (0, 0) とする), 行動として進行方向 $\theta(0 \leq \theta < 360[deg])$ を選択する. 行動を選択後, ロボットは選択した進行方向に 3[m/s] の速度で移動する. ロボットの目的はスタート (1, 30) からゴール ((84, 54) から半径 5[m] 以内) までの最短パスを学習することであり, ロボットがゴールに到着すると報酬 1 が与えられ, 再びスタート (1, 30) に戻される. その他の状態では報酬は 0 である. なお, ロボットが観測する状態 (x, y) と選択する行動 θ は共に連続値であり, ロボットは半径 0.5[m] の円形とした.

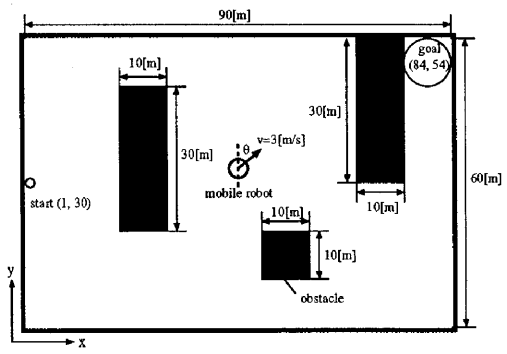

Fig. 3 The mobile robot simulation environment

4·2 actor-critic 法による強化学習 本実験では連続行動への拡張が容易である actor-critic 法と, 連続状態を扱うための tile-coding[8]を組み合わせた学習システムを採用する. その強化学習システムの概略図を図 4 に示す. エージェントがセンサによって観測した連続値データは, tile coding を介してバイナリデータに変換される. 強化学習アルゴリズムは actor-critic 法を採用し, 状態を表すバイナリデータは actor と critic のニューラルネットワーク (多層ニューラルネット, あるいはパーセプトロンでもよい) にそれぞれ入力される. actor は状態に応じて確率的に行動を選択する要素であり, ネットワークが出力する行動の平均値 $\mu(s_t)$ と標準偏差 $\sigma(s_t)$ に対応した正規分布に従って行動を選択する. 前述したように, そのような場合において critic によって望ましい行動であると判断された行動 a_t の選択確率を上げるには, 実行した行動値へ分布の中心 (平均値) を近づけ, 行動値が標準偏差の内側ならば分布の広がりを狭め, 外側ならば広げるように調整する[9]. ここでは, この原理を実現するために次のような簡単なアルゴリズムを作成した. まず, TD-error が正である場合, 次式を用いて正規分布の平均値を行動値に近づける.

$$\mu(s_t) \leftarrow \mu(s_t) + \alpha_\mu \delta_t (a_t - \mu(s_t)) \quad (12)$$

ここで, $\alpha_\mu(0<\alpha_\mu\leq 1)$ は $\mu(s_t)$ を出力するニューラルネットの学習率である. また, 次式を用いて正規分布の標準偏差を調整する.

$$\sigma(s_t)\leftarrow\sigma(s_t)+\alpha_\sigma\delta_t(target_\sigma-\sigma(s_t)) \quad (13)$$

ただし,

$$if \quad |a_t-\mu(s_t)|<\sigma(s_t), \ then$$
$$target_\sigma=(1-\beta)\sigma(s_t) \quad (14)$$
$$else$$
$$target_\sigma=(1+\beta)\sigma(s_t) \quad (15)$$

ここで, $\alpha_\sigma(0<\alpha_\sigma\leq 1)$ は $\sigma(s_t)$ を出力するニューラルネットの学習率, $\beta(0<\beta<1)$ は分布の広がりを変化させるパラメータである. このアルゴリズムでは, 行動 a_t と平均値 $\mu(s_t)$ の偏差が標準偏差 $\sigma(s_t)$ よりも小さい場合は $\sigma(s_t)$ が小さくなるように, 逆の場合は大きくなるようにネットワークを更新することになる. actor の更新後は, 式 4 を用いて critic の更新を行う.

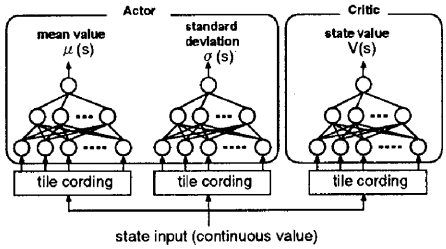

Fig. 4 The reinforcement learning system using actor-critic method

4.3 提案手法の適用法

4.3.1 教示データの作成と SOM の学習 まず, 教師が比較的良いと思われる状態-行動の組を予想し, 任意の数の教示データを作成するが, この問題では教示データは (x 座標, y 座標, 行動 (移動方向 [deg]) の形式で表すこととする. 教示データの内容を 5 に矢印で示す. 教示データは, 図 6 に示すように, それぞれ 2, 4, 8 与えた場合で実験を行った. なお, SOM のニューロンの数は $50\times 50=2500$, 発火ニューロンの学習率初期値を 0.9, 近傍ニューロンの学習率初期値を 0.8, 近傍領域の半径初期値を 40 とし, 共に時間の経過とともに減少させていった. また, SOM の学習回数は 5000 とした. 図 6(a), (b), (c) は, それぞれの教示データを用いて SOM に学習させ, 結果として得られた任意の状態の予想最適行動である. 各状態の予想最適行動は, 学習した SOM に状態 (x,y) を入力し, 発火したニューロンの行

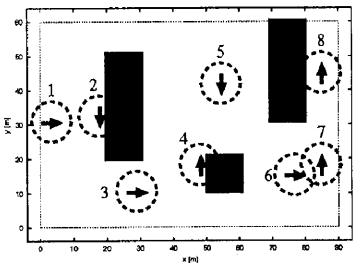

Fig. 5 8 teaching data directed by a human expert

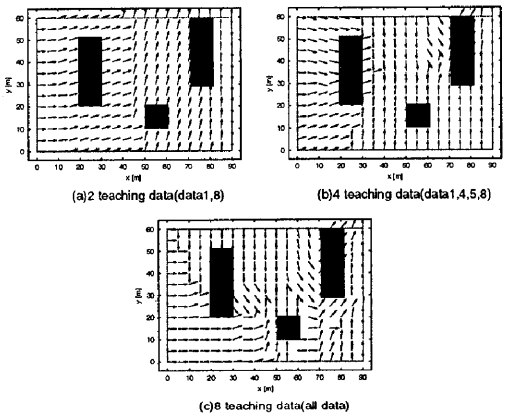

Fig. 6 Initial teaching data generated by the SOM

動を表す重みの値から得られる. これらの図から, SOM の学習によって教師が教示を行っていない状態でもある程度の最適行動が予想されていることがわかる. また, 教示データが多いほど実際の最適行動に近い行動が予想されていることもわかる. もちろん, 全ての予想された行動が最適であるわけではないため, 正確な最適政策を取得するためにはエージェントが試行錯誤によってこれを修正していくことが必要となる.

4.3.2 教示データの強化学習初期設定への反映 以上の手順によって得られた任意の状態 s の予想最適行動を参照して, actor の初期設定を行う. 具体的には, SOM の学習によって得られた予想最適行動の選択確率があらかじめ高く設定されるように, 任意の状態 s の平均値 $\mu(s)$ を予想最適行動に近づけ, 標準偏差を調整してそれらの行動の選択確率を高めておく. ただし, ここでは式 12,13 中の $\alpha_\mu\delta_t$ の代わりに, 事前学習の効果の度合いを示す bias-rate $b(0<b\leq 1)$ を用いる.

$$\mu(s)\leftarrow\mu(s)+b(a_{som}(s)-\mu(s)) \quad (16)$$

$$\sigma(s)\leftarrow\sigma(s)+b(target_\sigma-\sigma(s)) \quad (17)$$

ただし,

$$if \quad |a_{som}(s)-\mu(s)|<\sigma(s), \ then$$

$$target_\sigma = (1-\beta)\sigma(s) \qquad (18)$$

else

$$target_\sigma = (1+\beta)\sigma(s) \qquad (19)$$

ここで, $a_{som}(s)$ は SOM が出力した状態 s での予想最適行動である. bias-rate の値を大きくすると教示の効果が強まるが, 教示が誤っている場合に学習によって行動を修正することが困難になるため, b の値は比較的低めに設定してやるとよい. 前節の手順を踏まえ, SOM の学習が完了した後のアルゴリズムの詳細を図 7 に示す. 以上のアルゴリズムにより, タスクの実行に関して良い行動と予想される行動の選択確率が, 強化学習の初期段階において高く設定されるため, 強化学習の高速化が期待できる.

Repeat Z times:

(a) Select random state s

(b) Input s to the SOM and output the predicted optimal action $a_{SOM}(s)$

(c) Update mean value $\mu(s)$ and standard deviation $\sigma(s)$ (using (16)~(19))

Fig. 7 Pre-teaching algorithm (after learning of the SOM)

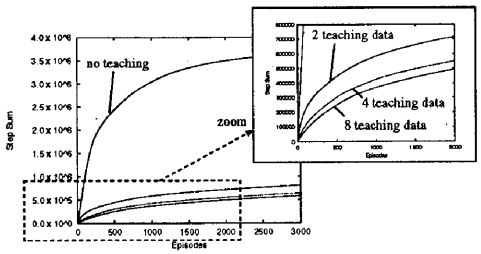

Fig. 8 Simulation results of the mobile robot navigation

4.4 シミュレーション結果

Table 1 Average steps in the early stage(episode1~100) and the closing stage(4901~5000) of reinforcement learning [steps / an episode]

	no teaching	2 data	4 data	8 data
episode 1~100	9930	2181	1117	777.7
episode 4901~5000	119.7	68.08	68.07	68.25

実験で用いたパラメータは, actor の学習率 $\alpha_\mu = \alpha_\sigma = 0.02$, critic の学習率 $\alpha_c = 0.02$, 割引率 $\gamma = 0.99$, 分布の広がりを調整するパラメータ $\beta = 0.1$, 事前学習の回数 $Z = 5000$, bias-rate $b = 0.01$ とした. また, actor と critic のニューラルネットワークのニューロンの重み初期値は-1~1 のランダム値であり, $\mu(s)$, $\sigma(s)$, 及び $V(s)$ の初期値はこれに依存する. 出力範囲はシグモイド関数の出力範囲である 0~1 である. tile cording の設定は, 5×5 の tiling を 5 枚重ね, 状態数は 885 となっている. 実験は教示データ数別にそれぞれ 100 回行い, 各エピソードごとのステップ数の平均値を算出して結果を比較した. 図 8 のグラフは, 横軸をエピソード数, 縦軸を累積ステップ数として, 教示数ごとの収束までに要するステップ数を比較したものである. グラフの傾きが一定の部分は学習が収束しているといえるので, 教示なしの場合と比較して, 教示ありの場合は約 2.5×10^6[step] 以上も収束までに要する行動回数が削減されていることがわかる. また, 教示数別に比較してみると, 教示数 2 と教示数 4 では約 2.0×10^5[step], 教示数 8 では教示数 4 よりもさらに約 0.5×10^5[step] の差があり, 教示数が多いほど少ない行動数で収束していることがわかる. また, 教示数が少ない場合でも十分な学習の高速化がなされていることから, 「少ない教示データで最大限の教示の効果が得られるようにする」という本研究の目的の 1 つが実現されていることも確認された. 次に, 教示によって学習初期段階にどの程度の行動回数が削減されているかを比較するため, 表 1 に 1~100 エピソードの平均ステップ数を示す. また, 教示が収束結果にどのような影響を及ぼすかを調べるため, 学習後半の 4901~5000 エピソードの平均ステップ数も表に記した. この表から, 学習初期段階では教示データ数が 2 の場合では教示なしの場合と比較して約 21%, 教示データ数が 4, 8 の場合ではそれぞれ約 11%, 8%に削減されていることが読みとれる. また, 教示を与えた場合の収束結果は, ほぼ等しいと判断できると思われるが, 教示を与えたほうが収束結果も良く, 最短経路を見つけやすいということがいえる.

Table 2 Average errors of actions generated by the SOM ($Error_{SOM}[deg]$) and actions generated by the actor before starting reinforcement learning ($Error_{actor}[deg]$)

	no teaching	2 data	4 data	8 data
$Error_{SOM}$	-	47.4	35.8	26.3
$Error_{actor}$	71.3	40.8	25.7	19.0

表 2 は,SOM が出力した教示データと強化学習後の actor の出力との誤差 ($Error_{SOM}$), 及び強化学習前の教示済 actor の出力と強化学習後の actor の出力との誤差 ($Error_{actor}$) を表にしたものである. これらのデータは,90[m] × 60[m] の環境内で x 軸,y 軸それぞれ 1[m] おきに誤差をとり (5,551 箇所), その平均を求めた.actor が出力する行動はロボットが進む方向であるため, 誤

差の単位は角度 [deg] となり，これらのデータは強化学習によってどの程度行動が修正されたかを示すものである．この表から，$Error_{actor}$ を比較すると，教示なしの修正角度と比較して，教示数 2,4,8 の場合でそれぞれ 57%,36%,27%の修正になっていることがわかる．また，$Error_{SOM}$ を比較すると，教示数が多いほど修正角が小さくなっていることから，今回の実験で使用した教示データはほぼ適切であったことがわかる．これらの結果は，教示を行うことにより，エージェントが学習によって行動を修正する度合を抑えることができ，その結果少ない試行回数で学習が収束することを裏付けるものであるといえる．

5. マルチエージェント追跡問題への適用

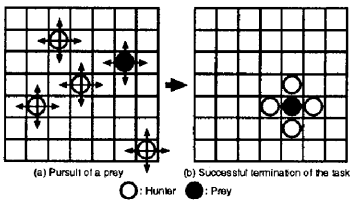

Fig. 9 Pursuit game

5·1 問題設定 本章では，マルチエージェントシステムのタスクとしてよく用いられる追跡問題 (pursuit game)[7]に提案手法を適用し，その有効性を検証する．以下に，追跡問題の概要を記す．図 9(a) に示すように，$n \times n$ の Gridworld 内に 4 台のハンター (hunter) と 1 台の獲物 (prey) が配置されている．制御対象となる各ハンターの目的は，ランダムな初期状態から獲物の追跡を開始し，図 9(b) に示すように獲物を囲い込むことである．各ハンターは互いに通信することなしに独立して行動し，単位時間ごとに上下左右のうち一方向へ 1 マス移動するか，その場に停止するかの 5 種類の行動の中から一つを選択する．また，獲物は単位時間ごとにそれらの 5 種類の行動をランダムに選択し続けて逃走する．ハンター同士，あるいはハンターと獲物は同時に同じ場所を共有することはできず，複数のエージェントが同じ場所に移動するような行動を選択した場合は，衝突したとして 1 ステップ前の場所に戻されるとする．ハンターの視界は深さ $d(2d+1<n)$ の範囲に限られており，視界範囲内に他のハンターあるいは獲物が存在する場合に限りそれらのエージェントと自分との相対座標を把握できる．また，視界中のエージェントがハンターか獲物かを区別できるとし，ハンターを知覚した場合はそのハンター番号 ($n_h = 1,2,3,4$) も認識できるとする．そして，図 9(b) のように獲物に隣接する四方のマスをハンターが占領したとき，獲物を捕獲したとして，各ハンターに報酬 1 が与えられる．このタスクを繰り返し，ハンターができるだけ短時間で獲物を捕獲できるような政策を取得するのが学習の目的である．なお，実験では，環境広さ 10×10，視界広さ $d = 3$ に設定した．

5·2 モジュール型強化学習の採用 ここで扱うようなマルチエージェント環境では，各エージェントは他のエージェントの位置を観測して共同作業を行わなくてはならないが，それぞれの学習エージェントの状態空間は環境内のエージェントの数に関係して指数関数的に増加してしまう．そこで本研究では，このような状態空間の爆発的増加を解決する有効な手法として提案されているモジュール型強化学習[7]を採用する．モジュール型強化学習では，学習エージェントは N 個の学習モジュール (learning module) と 1 つの調停モジュール (mediator module) を持つ．i 番目の学習モジュール L_i は，エージェントが時刻 t で観測した感覚入力 s_t のセンサ要素の中から自身が担当する感覚入力の部分集合 $x_i(t)$ を抽出し，状態 $x_i(t)$ における行動 $a_t \in A$ の評価値 $Q_i(x_i(t),a)$ を求める．そして，調停モジュールは以下の式を用いて状態 s_t における行動 $a_t \in A$ の評価値 $Q(s_t,a)$ を算出する．

$$Q(s_t,a) = \sum_{i=1}^{N} Q_i(x_i(t),a) \tag{20}$$

本実験では，i 番目の学習モジュール L_i が担当する感覚入力の部分集合 x_i は獲物と自分の相対座標 (x_p,y_p)，及び自分を除く i 番目のハンターと自分の相対座標 (x_{h_i},y_{h_i}) の 2 種類の座標で構成されることとする ($i = 1,2,3$)．従って，$N = 3$ となり，個々のハンターは 3 つの学習モジュールを持つことになる．また，ここではハンターの行動選択法として ε-greedy 法[8]を用い，$(1-\varepsilon)$ の確率で式 20 が最大になる行動を選択し，ε の確率でランダムに行動を選択することとする．ただし，$(0 \le \varepsilon < 1)$ である．

5·3 提案手法の適用法 追跡問題のような複雑なタスクでは，人間が最適な行動を完全に教示してやることは困難である．そこで，ここでは獲物の捕獲に部分的に有効であると予想される「獲物に近づく」行動をあらかじめ強化しておくことを試みる．時刻 t において，各ハンターのモジュール L_i には状態として 4 次元の座標 $(x_t^p,y_t^p,x_t^{h_i},y_t^{h_i})$ が入力されるが，図 10 に示すように獲物の座標を中心 (0,0) と考え，そこに近づく行動をあらかじめ強化する．10 は 8 つの教示データの作成例である．ただし，ここでは他のハンターとの相対座標 (x^{h_i},y^{h_i}) に関係なく獲物に近づく行動を強化する教示データを作成するため，例えば図 10 中の教示デー

タ1は, (0,1,#,#,*down*) というように#(Don't care) の記号を用いて表現する. 教示データを SOM に入力する際は, #の部分は入力の都度ランダムな値に設定して学習させる.

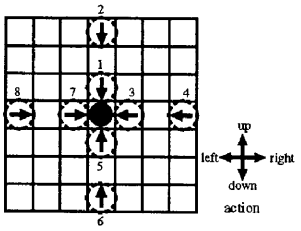

Fig. 10 8 teaching data for the pursuit game directed by a human expert

SOM の学習終了後は, その学習結果を用いて学習エージェントの Q 関数を初期化する. 初期化の手順は前章の実験の図7に示した手順と基本的に同じだが, ここでは学習アルゴリズムに Q-learning を用いるため, 図中の手順 (4) のみが異なる. 離散状態-離散行動を扱う Q-learning では, 一旦全ての Q 値を 0 に初期化した後,SOM によって予想された各状態における最適行動の Q 値のみ, 0 より大きい値 b に設定する. これにより, SOM によって予想された最適行動の学習初期段階における選択確率が高くなるため, 学習の高速化が期待できるが, 真の最適行動を獲得するためには適度な探索が必要となるため, b の値は比較的微少な値に設定する.

5·4 シミュレーション結果 実験では, 各学習パラメータは $\alpha = 0.05, \gamma = 0.9, \varepsilon = 0.05, b = 0.001$ とした. また, SOM のパラメータは前章の実験と同様の値に設定した. 実験は, 前節で解説した「獲物に近づく」行動の教示の他に, 群をつくって行動するように「味方のハンターに近づく」行動の教示も行って結果を比較した.

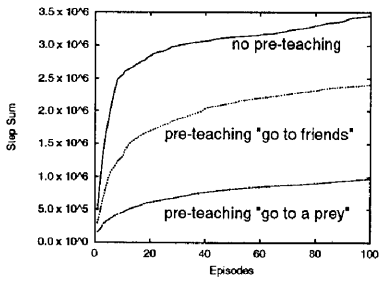

Fig. 11 Simulation results of pursuit games

図11は, 各エピソードごとの獲物捕獲までに要したステップ数の累積のグラフである. このグラフから, 獲物に近づくように事前学習を行った場合は教示なしの場合と比較して, 学習初期段階のステップ数が大幅に削減されており, より少ない試行回数で学習が収束していることがわかる. また, 味方のハンターに近づく行動を事前学習した場合も, 教示を行わなかった場合よりも収束までに要するステップ数がある程度削減されている. この結果から, 教示データとしては「獲物に近づく」行動を教示する方がより適切であったといえる.

6. おわりに

本論文では, 強化学習の収束を早めるために SOM を用いた事前教示法について新しい提案をした. 提案する手法は, 人間が事前に与える教示の数をできるだけ少ない代表値に限定し, それらの教示データを SOM に入力することで, SOM が事前教示のデータ数を自動的に拡大し, 結果として多くの事前教示を用いて強化学習を開始できるようにしたものである. この方法を用いることで, 人間が事前に与える教示データの数が少なくても SOM を援用することで, 強化学習のための事前教示の数を多くすることができ, 強化学習の初期段階において従来から問題となっていた無駄な試行錯誤が少なくなり, 学習の効率化が達成される. いくつかのシミュレーション結果から, 既存の教示法と比較して, 提案する手法が事前の教示の負担を軽減した有効な手法であることが明らかになった.

文献

(1) 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,(1997),811-821

(2) Lin,Long-Ji,Programming robots using reinforcement learning and teaching,Proceedings of the Ninth National Conference on Artificial Intelligence,(1991),781-786

(3) Clouse,J.A. and Utgoff,P.E.:A teaching method for reinforcement learning,Proceedings of the Ninth International Conference on machine learning,(1992),92-101

(4) Kaelbling,L.P.,Littman,M.L. and Moore,A.W.: Reinforcement Learning: A Survey,Journal of Artificial Intelligence Research 4,(1996),237-285

(5) Kohonen,T.:自己組織化マップ, シュプリンガーフェアラーク東京 (1996).

(6) Sehad,S. and Touzet,C:Neural Reinforcment Path Planning for the Miniature Robot Khepera,WCNN'95,Washington D.C.,USA,(1995),17-21

(7) Ono,N. and Fukumoto,K: A modular approach to multi-agent reinforcement learning,Distributed artificial intelligence meets machine learning,Springer-Verlag,(1997),25-39

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

(9) 木村元, 宮崎和光, 小林重信:強化学習システムの設計指針, 計測と制御,Vol.38,No.10,(1999),618-623