複数勝者 KFM 連想メモリを用いた強化学習の実現 (2)

池谷孝裕　長名優子

東京工科大学大学院 バイオ・情報メディア研究科コンピュータサイエンス専攻

1 はじめに

近年，自律分散型ロボットやマルチエージェントの研究が盛んになるにつれ，学習主体者が環境と相互作用し情報の獲得と行動を選択する学習方法として強化学習の研究が盛んに行われている．強化学習では Q-learning や Actor-Critic[1] など多くの手法が提案されている．また，ニューラルネットワークを用いて強化学習を実現するようなモデルも提案されているが，これらのモデルでは追加 (逐次) 学習ができないため，環境が変化すると学習を再度行わなくては適切な行動をとれないという問題がある．

本研究では，逐次学習が可能な複数勝者 KFM (Kohonen Feature Map) 連想メモリを用いて Actor-Critic による強化学習を実現する．

2 複数勝者 KFM 連想メモリを用いた強化学習

提案する複数勝者 KFM 連想メモリは，領域表現を用いた KFM 連想メモリ [2] を入力に類似した重みを持つ複数のニューロンが発火できるように拡張したモデルである．複数勝者 KFM 連想メモリは発火したマップ層のニューロンの内部状態の大きさに応じてそれらのニューロンの重みの値を重み付け和で出力を求める想起方法 (平均想起) と入力パターンに対応する重みの固定されたニューロンから 1 つを内部状態の大きさに応じた確率で選択し，そのニューロンの重みのみを用いて想起を行う想起方法 (1 対多想起) の 2 種類の想起方法のいずれかを用いて想起を行う．この特徴から複数勝者 KFM 連想メモリを強化学習に適用することにより，学習済みの環境に対してはそれに対応する行動を，未学習の環境に対しては既学習の内容をもとに適切な行動を出力することが可能となる．また強化学習において実際にエージェントが取る行動を複数勝者 KFM 連想メモリからの出力とランダムに決定された行動 (ランダム想起) から，最も目的に近づく行動を選択することによって試行錯誤が可能となる．

Realization of Reinforcement Learning using Multi-Winners KFM Associative Memory(2)
Takahiro Ikeya and Yuko Osana (Tokyo University of Technology, ikeya@osn.cs.teu.ac.jp, osana@cc.teu.ac.jp)

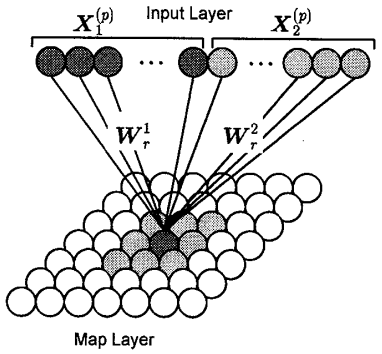


図 1: 複数勝者 KFM 連想メモリの構造

複数勝者 KFM 連想メモリは図 1 に示すように入出力層とマップ層の 2 層から構成されており，入出力層は複数のパターンを表す部分に分けられる．このモデルを強化学習に用いる場合には，入出力層は状態と行動を表す 2 つの部分に分けられる．

提案モデルにおいて強化学習は以下のような流れで行う．

(1) 複数勝者 KFM 連想メモリの重みを小さなランダムな値で初期化する．また，状態価値関数を 0 に初期化する．

(2) エージェントが環境 $s(t)$ を観測し，複数勝者 KFM 連想メモリによる行動の想起とランダム想起から，行動 $a(t)$ を決定する．行動の決定は以下のような手順で行う．

  (a) マップ層の各ニューロンについて内部状態を計算する．

  (b) 内部状態から勝ちニューロンを決定する．内部状態が閾値 θ^{map} 以上となるニューロンが勝ちニューロンの候補となる．候補のニューロンの中に重みの固定されているニューロンがある場合には 1 対多想起を行う．1 対多想起では候補となる重みの固定されているニューロンの中から 1 つのニューロンを内部状態の大きさに比例した確率で勝ちニューロンとして選択する．候補のニューロンの中に重みの固定されているニューロンがない場合には候補のニューロンすべてを勝ちニューロンとして，平均想起を行う．

(c) マップ層の内部状態をもとに出力を計算する．1対多想起では勝ちニューロンの重みを出力とする．また平均想起では，勝ちニューロンの内部状態の大きさに応じてそれらのニューロンの重みの値を重み付けして足し合わせたものを出力とする．

(d) 複数勝者 KFM 連想メモリによって出力された行動とランダム想起により出力された行動からより目的に近づくと思われる行動を実際の行動として決定する．

(3) エージェントが行動 $\boldsymbol{a}(t)$ を実行することにより，状態が $\boldsymbol{s}(t+1)$ に遷移する．

(4) クリティックは，環境の状態 $\boldsymbol{s}(t+1)$ から報酬 $r(t+1)$ を受け取り，TD 誤差 δ を出力する．

(5) TD 誤差 δ に基づき状態価値関数 $V(s(t))$ を更新する．

(6) TD 誤差に基づいて重みの更新を行う．

(I) TD 誤差が 0 より大きいとき

状態 $\boldsymbol{s}(t)$ に対して行動 $\boldsymbol{a}(t)$ をとったときにそれが望ましい行動であると判断された場合には，状態 $\boldsymbol{s}(t)$ と行動 $\boldsymbol{a}(t)$ からなる学習ベクトル $\boldsymbol{X}(t)$ を用いて学習を行う．ただし 1 対多想起により行動が決定された場合には状態 $\boldsymbol{s}(t)$ と行動 $\boldsymbol{a}(t)$ はすでに学習されていると考えられるので重みの更新は行わない．それに対し，平均想起やランダム想起によって行動が決定された場合には領域表現を用いた KFM 連想メモリ [2] と同様の学習アルゴリズムに基づいて重みを更新する．

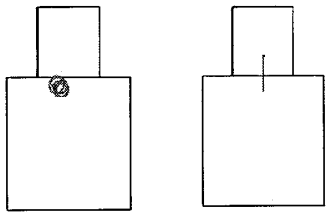
(II) TD 誤差が 0 より小さいとき

状態 $\boldsymbol{s}(t)$ に対して行動 $\boldsymbol{a}(t)$ をとったときに望ましくない行動であると判断されたときには，$\boldsymbol{X}(t)$ が想起されにくくなるように重みの更新を行う．また重みの更新を行うマップ層のニューロンが重みの固定されているニューロンである場合には，その固定を解除する．重みの修正方法は行動の決定に用いられた想起方法により異なる．1 対多想起により行動が決定された場合には行動の決定に用いた勝ちニューロンと入出力層の行動部分との間の重みを小さなランダムな値に変更する．平均想起で行動が決定された場合にはマップ層のニューロンの内部状態を計算し，内部状態が閾値 θ^m 以上となるマップ層のニューロンと入出力層の行動部分のニューロンとの間の重みを小さなランダムな値に変更する．ここで θ^m $(\theta^{map} \leq \theta^m)$ は重みの修正に関する内部状態に対する閾値である．またランダム想起により行動が決定された場合には，出力された行動は学習されていないため，重みの修正は行わない．

(III) TD 誤差が 0 のとき

TD 誤差が 0 のときには重みの更新は行わない．

(7) エージェントが目的を達成するまで (2)〜(6) を繰り返す．



(a) 学習前の軌跡　(b) 学習後の軌跡

図 2: 強化学習の結果

3 計算機実験

提案モデルを用いて車の車庫入れを例に実験を行った．車は前方にハンドルを一定の範囲の角度できりながら動くことができる．また，状態としては，車と駐車スペースの頂点の座標との距離，直前の動作，ハンドルの角度を用いる．

図 2 に提案手法での強化学習の結果を示す．この図では上のスペースが駐車スペースとなっており，図中の赤い線は車が移動した軌跡を示している．この結果より学習後車が駐車スペースにスムーズに向かっていることから提案手法において強化学習が行えることが確認できる．

参考文献

[1] I. H. Witten : "An adaptive optimal controller for discrete-time Markov environments," Information and Control, Vol.34, pp. 286–295, 1977.

[2] H. Abe and Y. Osana: "Kohonen feature map associative memory with area representation," Proceedings of IASTED Artificial Intelligence and Applications, Innsbruck, 2006.