事前知識を活用したMemory Reinforcement Learningによる行動獲得

Action acquisition by Memory Reinforcement Learning useing a prior knowledge

稲盛 有那 *1 Yuna Inamori　平川 翼 *1 Tsubasa Hirakawa　山下 隆義 *1 Takayoshi Yamashita　藤吉 弘亘 *1 Hironobu Fujiyoshi　柏原 良太 *2 Ryota Kashihara

稲葉 正樹 *2 Masaki Inaba　二反田 直己 *2 Naoki Nitanda

*1 中部大学 Chubu University　*2 株式会社デンソー DENSO CORPORATION

Obtaining a human-level control through reinforcement learning (RL) requires massive training. Furthermore, a deep learning-based RL method such as deep Q network (DQN) is difficult to obtain a stable control. In this paper, we propose a novel deep reinforcement learning method to learn stable controls efficiently. Our approach leverages the technique of experience replay and a replay buffer architecture. We manually create a desirable transition sequence and store the transition in the replay buffer at the beginning of training. This hand-crafted transition sequence enables us to avoid random action selections and optimum local policy. Experimental results on a lane-changing task of autonomous driving show that the proposed method can efficiently acquire a stable control.

1. はじめに

Alpha Go[1] が囲碁の世界チャンピオンに勝利して以降，様々な分野で強化学習の活用が期待されている．強化学習は，一般的な教師あり学習と異なり，環境とエージェントのやりとりで得られる報酬を手掛かりに学習する手法であり，未知の環境でも学習が行えるというメリットがある．強化学習は車両やロボットの自律制御 [2] や物体把持 [3] など様々なタスクに応用されている．近年では，Deep Q-Network(DQN)[4] をはじめとする深層強化学習により，複雑なゲームや制御問題を解決することが可能となっている．

深層強化学習による行動獲得は高い性能を示す一方で，複雑な問題に対する学習が難しいという問題が存在する．例えば，DQN のようなニューラルネットワークを用いる手法では，状態を連続値としてネットワークに入力しネットワークの演算を経て行動を出力する．この時，多層のネットワークを用いることで，複雑な問題に対する制御を獲得することが可能となる．しかし，このような多層のネットワークを学習する場合，学習するパラメータ数の増加や勾配消失等により学習が困難となる．また，パラメータはランダムに初期化され学習を開始するが，これはランダムな行動選択から学習を行うことに等しい．特に，時系列行動を強化学習によって獲得する場合，現在の状態に対する制御が過去の制御に依存するため，離散的な行動に比べて困難となる．DQN における学習を安定化させる手法として，過去の行動遷移を蓄積し，蓄積したデータを繰り返しランダムサンプリングして利用する experience replay がある．しかし，ランダムな初期値からの学習では，学習に寄与しない行動選択を蓄積するため，最適な行動を選択するまでに多くの時間を要する．さらに，大規模な状態空間や複雑なタスクに対する学習では局所解に陥りやすく，望ましい行動を獲得することが難しい．

画像認識における Convolutional Neural Network(CNN) では，上記の問題を回避し高い認識性能を実現するために，ImageNet[5] などの大規模な一般物体認識のためのデータセットを用いて学習したモデル (事前学習モデル) のパラメータを初期値とし，個々のタスクに対して学習を行う転移学習が広く用いられている．この事前学習モデルの活用は，タスクが異なった場合でも画像中の特徴が類似していると仮定した上で用いられている．しかし，深層強化学習では問題に応じてその状態空間および入力が異なるため，事前学習モデルの導入が困難である．

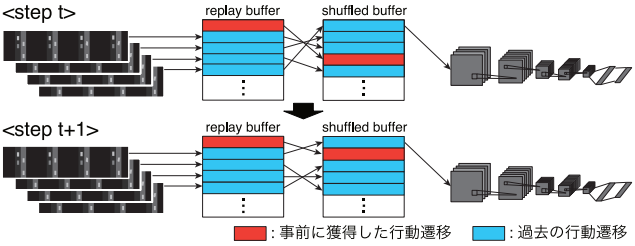
そこで本稿では，事前学習モデルではなく，安定した行動に対する事前知識を学習に導入することで，上記の問題を抑制し，効率的に安定した制御を獲得するための Memory Reinforcement Learning を提案する．提案手法では，事前に獲得した行動遷移を experience replay の replay buffer にあらかじめ保存し，学習を行う．学習時に，replay buffer 内の主導で作成した行動遷移が活用されることで，学習の初期から安定した行動を生成し，効率的に安定した制御を獲得することが可能である．

自動運転車両の車線変更を行うシミュレータを用いた実験にて，提案手法を用いることで安定した動作を効率的に獲得できることを示す．また，replay buffer 内の一部を常に事前知識にして学習した場合および，事前知識のみで学習を行った後に通常の学習を行う場合の性能比較および，事前知識と報酬の組み合わせによる精度の変化について確認する．

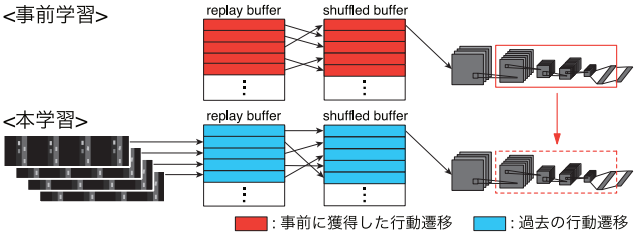
2. 関連研究

強化学習において，効率的に安定した制御を獲得する手法はいくつか提案されている．Peng ら [6] は二足歩行の動作獲得を目的とし，階層的な強化学習手法を提案している．この手法では，目的地まで到達するための長期的な制御と歩行動作を制御する短期的な制御を行う 2 種類の制御システムを用いるこ

連絡先:
稲盛 有那：yuna@mprg.cs.chubu.ac.jp
平川 翼：hirakawa@mprg.cs.chubu.ac.jp
山下 隆義：yamashita@cs.chubu.ac.jp
藤吉 弘亘：hf@cs.chubu.ac.jp
柏原 良太：RYOTA_KASHIHARA@denso.co.jp
稲葉 正樹：MASAKI_INABA@denso.co.jp
二反田 直己：NAOKI_NITANDA@denso.co.jp



(a) 事前知識 A



(b) 事前知識 B

図 1: 提案手法での事前知識の与え方

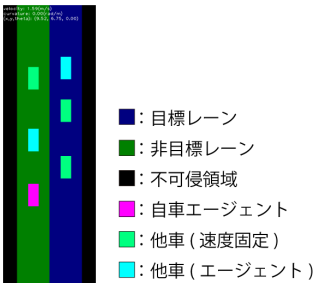


図 2: 環境画像例

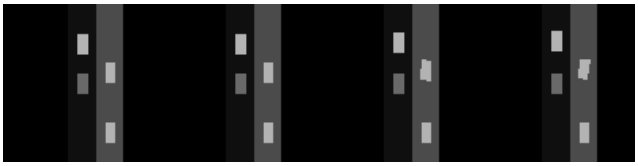


図 3: 入力画像例

とで，複雑な地形や環境に対応した歩行動作を獲得している．一方，提案手法では事前に獲得した安定した行動遷移を学習開始時から活用することで，安定した動作を獲得する．

高次元な状態空間における行動を獲得するために，Memory Network[7] を用いて過去の行動を効率的に活用する手法が提案されている [8]．Parisotto ら [9] は長期にわたる過去の記憶を保存し学習に用いるために，Neural Map と呼ばれる手法を提案している．この手法は，3D 環境における強化学習エージェント専用に設計された構造化メモリである．適応可能な書き込み操作を使用しているため，メモリサイズと計算コストは時間の経過と共に増加しない．また，書き込み操作に特定の帰納バイアスを課しており，ナビゲーションが正常な動作の中核要素である 3D 環境に適している．一方，提案手法では単純な過去の行動ではなく，事前に獲得した安定した行動を replay buffer に記憶し学習を行う．

また，事前に獲得した行動を活用する方法として，模倣学習 [10] を導入した手法が存在する [11]．まず模倣学習で他社の行動を真似ることで大まかに望ましい行動を獲得し，その後副報酬を伴う強化学習を行う．模倣学習で得た方策を強化学習による試行錯誤で修正することで，最終的に望ましい行動を獲得する．しかし，模倣学習による行動獲得そのものが難しい問題であり，安定した動作が獲得できなければその後の学習が困難になるという問題が考えられる．一方，提案手法では安定した行動遷移を事前知識として用いるため，不安定な動作の獲得を回避することが可能である．

3. 提案手法

本節では，提案手法について述べる．前述の通り，深層強化学習において安定的な制御を効率よく獲得するために，experience replay を活用する．これは，試行により獲得した状態，行動，報酬のデータを Replay buffer と呼ぶメモリに蓄積する．そして，学習する際に蓄積したデータを繰り返しランダムサンプリングして利用する．通常の強化学習では，学習開始時には Replay buffer の中身は空であるため，一定のエピソードを経て Replay buffer にデータを蓄積させてから学習が始まる．また，初期の Replay buffer 内のデータは殆どランダムな行動をした時のデータである．そのため，膨大な状態と行動の組み合わせから最適な行動を学習するのに時間がかかる．そこで提案手法では，事前知識として望ましい行動遷移を手動で作成し，replay buffer へ格納し，それをもとに学習を開始する．あらかじめ replay buffer に事前知識を蓄積しておくことで，学習の初期から事前知識を用いて学習されるためランダムな行動選択を抑制できる．これにより，効率よく学習を行うことができる．本稿では，作成した事前知識の与え方を 2 種類提案する．図 1 に，事前知識の与え方のイメージを示す．図 1 の replay buffer，shuffled の中の赤色は事前知識，青色は学習中に得た行動遷移を表す．

3.1 事前知識 A

図 1(a) に示すように，replay buffer 内の 1 割に事前知識を常に蓄積しておく．残りの 9 割では従来の replay buffer と同様に，学習が進むにつれデータが入れ替わっていく．これにより，repray buffer 内には常に望ましい行動遷移が蓄積されているため，一定の割合で事前知識を用いた学習を行い，ランダムな行動選択を抑制することが可能となる．

3.2 事前知識 B

はじめに replay buffer に事前知識として作成した行動遷移データを蓄積しておく．この状態で深層強化学習を行う．この処理を事前学習とする．その後に本学習として，学習したモデルを初期値にして従来の深層強化学習を行う．これにより，事前知識をもとに獲得した行動から，さらに最適な行動を獲得することが期待できる．

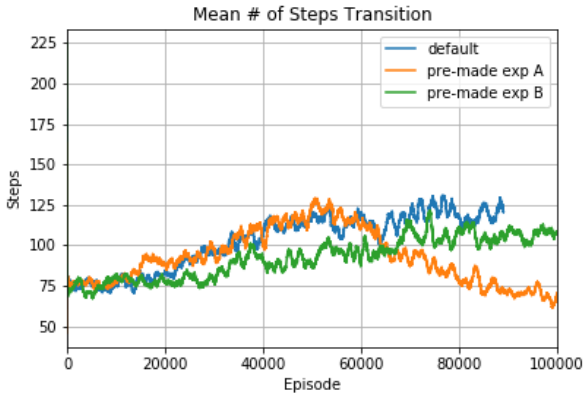
4. 評価実験

事前知識の有効性を評価するためのタスクとして，車線変更タスクを用いる．車線変更タスクでは，現在の走行レーンから隣の走行レーンへ移動する．その際，他車が複数台おり，衝突を回避しながら移動しなければならない．また，明確なゴール地点が存在せず，長期の行動を獲得することが必要なタスクである．そのため，事前知識なくランダムな行動から最適な行動を獲得することが困難なタスクとなっている．

4.1 実験概要

図 2 に，車線変更タスクのシミュレータ環境を示す．青色は目標レーン，緑色は非目標レーン，黒色は不可侵領域であり，ピンク色は自車エージェント，黄緑色と水色は他車とする．他車の総数は，学習時は全エピソードで 5 台固定，テスト時はエ

(a) 平均収益



(b) 平均生存ステップ

図 4: 各エピソードにおける平均収益および平均生存ステップ数の推移

ピソード毎に 0〜5 台のランダムで行う．他車は，固定速度で直進する黄緑色のものと，自車エージェントのモデルをコピーして作られる水色のものの 2 種類存在する．他車が一定速度，または実車と同じエージェントであると，タスクとして簡単になるが，実環境ではドライバの運転スキルにより行動が異なる．そのような環境を想定し，他車に自車エージェントの過去のモデルをランダムに選択して用いる．環境を表した画像から，自車周辺を 84×84 ピクセルに切り出したグレースケール画像 4 フレームを入力とする．図 3 に，入力画像の例を示す．エージェントが可能な行動は，速度 3 種，曲率 3 種を掛け合わせた 9 つである．速度は，± 0m/s（速度維持），$+0.1$m/s(加速)，$-.0.1$m/s(減速) の 3 種である．曲率は，0(車体角度維持)，$+0.01$rad/m(左回転)，-0.01rad/m(右回転) の 3 種である．これらは，実環境での動作を想定し，実際の車両で制御可能な範囲である．

本実験では，強化学習法として Double DQN[12] を用いる．従来の Double DQN で学習したものをデフォルトとし，事前知識 A，事前知識 B と比較を行う．表 1 に学習条件，表 2 に学習に用いる CNN の構造を示す．

4.2 実験結果

図 4 に，学習時の平均収益，平均生存ステップ数の推移を示す．事前知識 A は，平均収益が 5 万エピソードをピークに途中から低下している．また，平均生存ステップ数も同様に 5 万エピソード付近から低下している．10 万エピソード時点では事前知識 A が最もスコアが低い結果となった．一方，事前知識 B は，平均収益が低下することなく学習できている．また，平均生存ステップ数も同様に低下することなく，増加していることが分かる．

また，表 3 にテストスコアを示す．テスト回数は 100 回であり，成功時に 1 点として計測している．初期レーンは，スタート時の自車エージェントの位置を表す．初期レーンがランダムの場合，初期レーンが必ず非目標レーンである場合の 2 通りでテストを行った．生存スコアは 300 ステップ到達時までに自車が他車や壁に衝突しなかった回数，目標到達スコアは 300 ステップ到達時に自車が車線変更し，目標レーンに達した回数である．人の項目のスコアは，6 名に 100 回ずつテストプレイを行ってもらったスコアの平均である．事前知識を用いないデフォルトは学習後期において，収益，ステップ数ともに最も高くなっている．しかしながら，表 3 に示すようにテストスコアでは事前知識 B よりも低くなっている．これは，最適ではない局所解に陥ったためと考えられる．一方で，デフォルト

表 1: 実験での学習条件

報酬	他車または壁に衝突：-5 衝突していない場合は下記を合算 車線中央付近：+1 停止：-1 目標車線：+1 非目標車線：-2
エピソード終了条件	衝突または 300 ステップ経過
学習エピソード数	最大 100000 エピソード
割引率	0.95
探索法	Linaer decay epsilon greedy
Replay buffer サイズ	$1e+5$

表 2: CNN の構造

	処理	詳細
入力層	サイズ	$4 \times 84 \times 84$
畳み込み層 1	活性化関数	ReLU
	フィルタ	$3 \times 3 \times 16$
	maxpooling	2×2
畳み込み層 2	活性化関数	ReLU
	フィルタ	$3 \times 3 \times 16$
	maxpooling	2×2
畳み込み層 3	活性化関数	ReLU
	フィルタ	$3 \times 3 \times 16$
	maxpooling	2×2
全結合層	活性化関数	ReLU
	ユニット数	256
出力層	ユニット数	9

と比較して，事前知識 A は生存スコアが同等であるが，目標到達スコアが低い．これは，事前知識として与えた経験に頼りすぎ，新たな経験の獲得が十分にされていないことが考えられる．事前知識 B はデフォルトおよび事前知識 A と比べて，高いスコアとなっている．事前知識 B では学習初期は事前知識のみで，学習が進むにつれ replay buffer 内は新しい経験に入れ替わる．しかし，初期で既に人に与えられた行動遷移を得ており，ランダムに参照を行なった場合でも全てに事前知識が導入されているため，さらによい行動を獲得し，テストスコアが上昇したと予想される．したがって，事前知識からタスクで実行するのに有効な基本行動を獲得し，それをもとにさらにタスクに適した行動を獲得できていると考える．

また，人のスコアと比較しても，事前知識 B は生存スコアは約 32%，目標到達スコアは約 31%高い結果となっている．このことから，事前知識 B の手法は車線変更に伴う車体制御が

表 3: テストスコアの比較

	初期レーン	生存スコア	目標到達スコア
デフォルト	ランダム	53	50
	非目標	33	29
事前知識 A	ランダム	49	37
	非目標	37	10
事前知識 B	ランダム	**67**	**66**
	非目標	**51**	**48**
人	ランダム	50.7	50.3

安定しており，効果的であると言える．

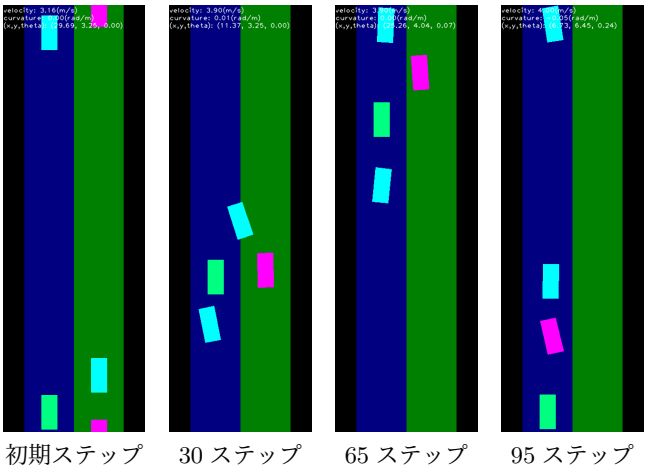
4.3 考察

本実験でテストスコアが最も高かった事前知識 B について，テストシーンにおける動作例を図 5 に示す．図 5(a) では，他車が 3 台いるシーンにおいて，スタート時に自車の前方を走行する他車が車線変更を試みている．その際，自車は他車との衝突を避けるために加速せずに追従している．そして，他車が車線変更後，他車間の間隔が十分に空いたところにスムーズに入る行動を取ることが出来ている．また，図 5(b) では，他車が 3 台おり，隣の車線に 2 台走行している．このようなシーンにおいて，自車は加速して目標車線にいる他車を追い抜き，充分な車間距離になってから車線変更を試みている．このように他車との速度と車間距離を考慮することが出来ている．これらの結果より，提案手法では，人が実環境で車線変更を行う際に取りうる動作を深層強化学習により獲得することができた．

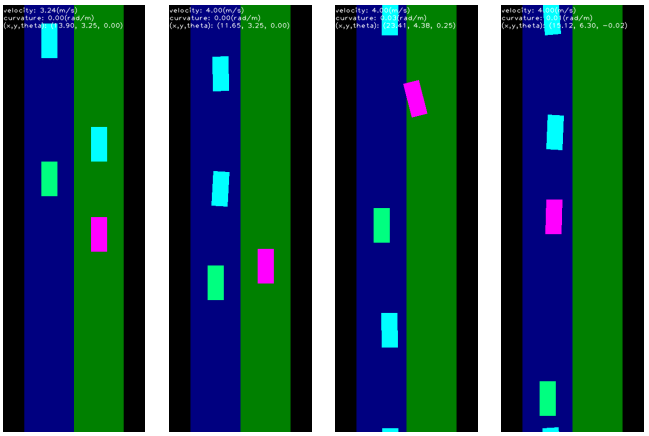
5. おわりに

本稿では，深層強化学習において効率的に安定した制御を獲得するための Memory Reinforcement Learning を提案した．提案手法では，experience replay の replay buffer を活用し，事前に獲得した行動遷移を replay buffer に保存し学習を開始する．これにより，学習初期のランダムな行動選択を抑制し，不安定な行動の獲得を回避することが可能である．自動運転車両の車線変更タスクを用いた実験にて，事前に獲得した行動遷移を用いずに学習した場合と比較し，安定した制御を効率的に獲得できることを確認した．

今後の課題として，事前知識を効果的に活用する方法の検討や事前知識を踏まえた報酬の設計などが挙げられる．



(a) 他車に追従してから車線変更する例



(b) 他車を追い抜き車線変更する例

図 5: 事前知識 B のテストシーンにおける動作例，青:目標レーン，緑色:非目標レーン，黒:不可侵領域，ピンク:自車，黄緑・水色:他車を表している．

参考文献

[1] D. Silver, et al., “Mastering the game of Go with deep neural networks and tree search,” Nature, Vol. 529, pp. 484–489, 2016.

[2] T.P. Lillicrap, et al., “Continuous control with deep reinforcement learning,” In ICLR, 2016.

[3] L.Pinto, *et al*. Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection. In IEEE, 2016.

[4] V. Mnih, *et al*. Human-level control through deep reinforcement learning. Nature 518, pp.529–533, 2015.

[5] J. Deng, et al., “ImageNet: A large-scale hierarchical image database,” In CVPR, 2009.

[6] X.B.Peng, *et al*. DeepLoco: Dynamic Locomotion Skills Using Hierarchical Deep Reinforcement Learning. ACM Transactions on Graphics, Vol. 36, No. 4, Article 41, 2017.

[7] S. Sukhbaater, et al., “End-to-end memory networks,” In NIPS, 2015.

[8] J. Oh, et al., “Control of Memory, Active Perception, and Action in Minecraft,” In ICML, 2016.

[9] E.Parisotto, et al., “Neural Map: Structured Memory for Deep Reinforcement Learning,” In ICLR, 2018.

[10] E. Parisotto, et al., “Imitation Learning: A Survey of Learning Methods,” ACM Computing Surveys, Vol. 50, No. 2, pp. 21:1–21:35, 2017.

[11] 田渕一真ら, “模倣学習と強化学習の調和による効率的行動獲得,” 人工知能学会全国大会論文集, pp. 212–215, 2016.

[12] H. Hassel, et al., “Deep Reinforcement Learning with Double Q-Learning,” In AAAI, 2016.