1A2-OS-05b-1

深層強化学習におけるオフライン事前学習法

An Offline Pretraining Method for Deep Reinforcement Learning

那須野薫 *1　　松尾豊 *1
Kaoru Nasuno　　Yutaka Matsuo

*1東京大学
The University of Tokyo

Deep reinforcement learning has been gaining more and attetion recently. Research on deep reinforcement learning methods claims their performance such as learnability from scratch, stability of learning and generalization performance on a variey of tasks. However, these research mainly focuses on effectiveness of the policy learned online but not offline. In some practical situation, we can utilize near-optimal trajectory by human that can be used for offline pretraining. This paper will empirically analyze the policy of one of the recently proposed deep reinforcement learning method *Normalized Advantage Function* or *NAF* pretrained offline with trajectory gathered by near-optimal policy. In our experiments, we will show that the policy pretrained by offline Q-learning after supervised learning converge faster than the policy trained from scratch, the policy pretrained by offline Q-learning and the policy pretrained by supervised learning.

1. はじめに

近年，深層強化学習の研究の進展は目覚ましい．ゲームプレイ [Mnih 13]，ロボティクス [Levine 16a]，囲碁の勝負 [Silver 16] 等と幅広い領域で研究成果が報告されている．また，さまざまなタスクで方策の学習が可能な汎化性の高い手法 [Schulman 15, Lillicrap 15, Gu 16] が多く提案されている．

こうした深層強化学習の研究報告は，タスクへの新しい適用法 [Silver 16] や，収束までのエピソード数 [Gu 16]，収束の安定性 [Schulman 15] を主張するものが多い．そのため，こうした報告では，方策関数をゼロから学習させる場合の議論が行われることが多く，逆に，既存のログデータを用いたオフラインでの学習方法や学習効果ついての議論は多くない．

さて，これまで，自動車の操作や化学プラント機械の操作は，主に人間のオペレータによって行われてきた．近年，人口減少や少子高齢化により，このような機器の操作を担う次世代のオペレータの不足が懸念されている．人口減少や少子高齢化は，人口構造に基づいた問題であり，解決は容易ではない．そのため，産業全体の労働力減少や，その結果としての生産力や競争力の低下が危惧されている．

こうした労働人口減少の影響の緩和策として，機械操作の自動化が期待されている．機械操作の自動化は，これまで多く試みられてきた．しかし，技術開発が追い付いておらず，課題が顕在化しつつある上記の領域への導入は限定的である．機械操作は，ハードウェアとソフトウェアの両方から成り立っており，そのため，その自動化には両方の性能の向上が必要であると考えられる．そのうち，ソフトウェアの性能向上の枠組みとして注目されはじめたのが，深層強化学習である．深層強化学習による機械の自動操縦は，こうした労働力減少を補い，業界を支える可能性として期待されている．

このような機械の操作の自動化を目指し，深層強化学習を適用する場合，オンラインの強化学習によって一から方策を学習させる必要は必ずしもない可能性がある．自動車の操作や化学プラント機械の操作等においては，人間のオペレータが状況に応じて行った操作をログデータとして記録できる．この操作ログは，人間のオペレータが一定以上の水準で行った操作ログであり，強化学習モデルをオンラインで適用させる前の事前学習に活用できる可能性がある．

また，事前学習は，実環境適用の際の危険性を低減させる可能性がある．深層強化学習は，方策関数の収束にオンラインで多くの試行回数が必要である．収束前の方策は，有効ではない可能性が高く，実環境への適用は危険が伴う可能性がある．したがって，方策関数の収束を早め，実環境適用の際の危険性を低減させる可能性があるオフラインでの学習は，実応用を検討する場合，重要である．

そこで，本研究では，深層強化学習のモデルをいくつかの方法によって事前にオフラインで学習させ，その効果を分析する．深層強化学習の手法は，近年提案された行動空間が連続値であり方策オフ型の手法である Normalized Advantage Function (NAF)[Gu 16] を用いる．オフラインでの事前学習は，A) 方策関数のみを教師あり学習させる，B) オフラインで強化学習させる C) 方策関数のみを教師あり学習させたのち，オフラインで強化学習させる D) 方策関数のみを教師あり学習させたのち，オフラインで方策関数以外を強化学習させるの 4 つの方法を適用する．実験は，OpenAI gym[Brockman 16] で提供される 2 種類のシミュレータを対象に行う．実験結果の章では，C) 方策関数のみを教師あり学習させたのち，オフラインで強化学習させる事前学習法を適用した結果が最も収束の安定性が高いことを述べる．

2. 関連研究

本章では，複数ある深層強化学習の手法のうち，本研究で，Normalized Advantage Function (NAF)[Gu 16] を用いる理由について述べ，その後，NAF の仕組みについて述べる．

深層強化学習の手法を分類する方法はいくつかある．例えば，適用先が主にゲームやシミュレータである手法 [Mnih 13, Silver 16] か，実世界のロボットや機械である手法 [Levine 16a, Levine 16b] かという分類や，方策オフ型の手法 [Lillicrap 15, Gu 16] か，方策オン型の手法 [Schulman 15] かという分類がある．また，行動空間が離散値の手法 [Mnih 13] か連続値の手法 [Gu 16] かという分け方もある．

連絡先: 那須野薫，東京大学，nasuno@weblab.t.u-tokyo.ac.jp

さて，実世界への応用という観点では，望ましい手法の区分があると考えられる．例えば，実世界では，方策の 1 回の適用にシミュレータより多くの時間やコストがかかることが多い．そのため，主にシミュレータやゲームへの適用を前提としている手法のように多くの試行回数を前提とする手法の実世界への適用は難しい．また，自動車のハンドルや化学プラント機械のレバー等，実世界の行動空間は連続値のものも多くあり，行動空間が離散値であることを前提とした手法の適用は限定的であると考えられる．さらに，人間のオペレータの操作履歴を活用する場合，方策オン型の手法は，学習に用いるデータがそのデータの生成に用いた方策であることを前提とするため，活用できない．これらを踏まえると，実世界への応用という観点では，収束までの試行回数がそれほど多くなく，行動空間が連続値であり，方策オフ型の手法であるものが望ましい手法の区分であると考えられる．

この望ましい区分に含まれる手法として，Deep Deterministic Policy Gradient (DDPG)[Lillicrap 15] や Normalized Advantage Function (NAF)[Gu 16] という手法がある．DDPG は Actor-Critic のアプローチに基づいた手法であり，NAF は Q-learning のアプローチに基づいた手法である．[Gu 16] では，NAF は DDPG よりも多くのタスクにおいて，収束までにかかるエピソード数が少ないことが主張されている．どちらの手法も多少の長所短所はあるものの，本研究では，この点に着目して DDPG ではなく NAF を用いる．

2.1 Normalized Advantage Function

Algorithm 1 Normalized Advantage Function の処理の流れ

1: Q 関数のランダム初期化 $Q(\mathbf{s}, \mathbf{a} \mid \theta^Q)$
2: ターゲットネットワーク $Q^{'}$ の初期化 $\theta^{Q^{'}} \leftarrow \theta^Q$
3: リプレイバッファ R の初期化 $R \leftarrow \emptyset$
4: **for** $episode = 1, max\_episode$ **do**
5: 　ノイズプロセス N の初期化
6: 　初期状態を観測 $\mathbf{s}_1 \sim p(\mathbf{s}_1)$
7: 　**for** $t = 1, max\_timestep$ **do**
8: 　　行動 $\mathbf{a}_t$ の選択 $\mathbf{a}_t = \mu(\mathbf{s}_t) + N$
9: 　　行動 $\mathbf{a}_t$ の実行
10: 　　遷移 $(\mathbf{s}_t, \mathbf{a}_t, r_t, t_t, \mathbf{s}_{t+1})$ を R に追加
11: 　　R より，バッチサイズ分の遷移データをランダムサンプリング
12: 　　$y_i = r_i + \gamma V^{'}(\mathbf{s}_{i+1} \mid \theta^{Q^{'}})$ を代入
13: 　　ロス関数 $L = 1/N \sum_i (y_i - Q(\mathbf{s}_i, \mathbf{a}_i \mid \theta^Q))^2$ が最小となるようにパラメタ θ^Q を更新
14: 　　ターゲットネットワークの更新：$\theta^{Q^{'}} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q^{'}}$
15: 　**end for**
16: **end for**

NAF の処理の流れを Algorithm 1 に示す．$\mathbf{s}_t, \mathbf{a}_t, r_t, t_t$ はそれぞれ，タイムステップ t における状態，行動，その結果得られた報酬と，エピソードの終端か否かを示す．Q は Q 関数全体を，$Q^{'}$ はそのターゲットネットワークを，μ は方策関数を，R は過去ログデータを蓄積するリプレイバッファを指す．Q 関数は，$Q(\mathbf{s}, \mathbf{a}) = A(\mathbf{s}, \mathbf{a}) + V(\mathbf{s})$ とアドバンテージ関数とバリュー関数で表現され，アドバンテージ関数は $A(\mathbf{s}, \mathbf{a}) = -1/2(\mathbf{a} - \mu(\mathbf{s})) \cdot L(\mathbf{s}) \cdot (\mathbf{a} - \mu(\mathbf{s}))$ と，正規化された二次形式で表現される．これにより，アドバンテージ関数が最大となり，結果，累積期待報酬が最大となる行動は $\mathbf{a} = \mu(\mathbf{s})$ となる．

なお，本研究では，[Gu 16] とは異なり，Imagination Rollouts は用いない．Imagination Rollouts はサンプル効率を向上させるために提案された仕組みである．本研究は，過去ログデータの活用による学習の効率性について議論するものである．近い目的を達成するための Imagination Rollouts を用いると議論が曖昧となるため，本研究では用いない．

[Gu 16] では，Q 関数はランダムに，リプレイバッファは空にして初期化が行われていた．しかし，必ずしも Q 関数の初期化はランダムに行う必要はなく，また，リプレイバッファも必ずしも空が最適であるとは限らない．例えば，高い報酬が得られたエピソードの状態と行動のペアがあれば，方策関数はそのデータを元に教師あり学習をしたほうがいいかもしれない．また，Q 関数も過去の事例に基づいてある程度学習させたほうがいいかもしれない．本研究では，こうした初期化を工夫するものとも解釈できる．

3. オフラインでの事前学習法

事前学習の方法は実環境への応用を想定して設計したい．人間のオペレータが一定以上の水準で行った操作ログデータを元に事前に学習させることを想定する場合，学習させる方法やパラメタによっていくつかバリエーションがある．

想定の利用可能なログデータは一定以上の水準のものであり，ランダムよりは見習うべき操作履歴であると考えられる．したがって，教師あり学習によって方策関数を事前学習することは有効そうである．また，結果の良否を無視すれば，任意のログデータからデータをサンプリングして，強化学習によってネットワーク全体を事前学習することは手続きとしては可能である．加えて，その場合，教師あり学習した方策関数を再学習すべきか否かという論点もある．本研究では，特に，NAF における方策の事前学習に関する下記の 4 つのバリエーションについて検証する．リプレイバッファの初期化については，今後の課題としたい．

A 方策関数のみを教師あり学習させる．
対象データよりバッチサイズごとにサンプリングしたデータに対してロス関数 $L = \sum(\mathbf{a} - \mu(\mathbf{s}))^2$ を最小化させるように θ^μ を学習させる．

B オフラインで強化学習させる．
対象データよりバッチサイズごとにサンプリングしたデータに対して Algorithm1 の 13, 14 行にしたがって，$\theta^Q, \theta^{Q^{'}}$ を学習させる．

C 方策関数のみを教師あり学習させたのち，オフラインで強化学習させる．
対象データよりバッチサイズごとにサンプリングしたデータに対してロス関数 $L = \sum(\mathbf{a} - \mu(\mathbf{s}))^2$ を最小化させるように θ^μ を学習させる．その後，Algorithm1 の 13, 14 行にしたがって，$\theta^Q, \theta^{Q^{'}}$ を学習させる．

D 方策関数のみを教師あり学習させたのち，オフラインで方策関数以外を強化学習させる．
対象データよりバッチサイズごとにサンプリングしたデータに対してロス関数 $L = \sum(\mathbf{a} - \mu(\mathbf{s}))^2$ を最小化させるように θ^μ を学習させる．その後，Algorithm1 の 13, 14 行にしたがって，$\theta^Q, \theta^{Q^{'}}$ を学習させる．その際，$\theta^\mu, \theta^{\mu^{'}}$ は学習させない．

表 1: タスクおよび学習方法ごとの累積報酬

タスク	事前学習に用いるレコード数	事前学習なし	事前学習法 A	事前学習法 B	事前学習法 C	事前学習法 D
Pendulum	10,000	−744 ± 528	−442 ± 510	−502 ± 502	−504 ± 514	−627 ± 594
	30,000		−522 ± 531	−622 ± 475	**−294 ± 379**	−673 ± 504
	100,000		−502 ± 523	−916 ± 469	−413 ± 439	−776 ± 565
Mountain Car	10,000	87 ± 6	90 ± 6	90 ± 4	90 ± 4	**93 ± 1**
	30,000		79 ± 35	87 ± 10	83 ± 23	91 ± 5
	100,000		**93 ± 1**	82 ± 15	**93 ± 1**	92 ± 1

図 1: タスク Pendulum のイメージ

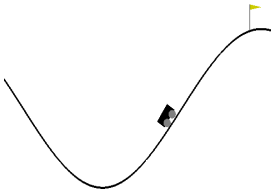

図 2: タスク Mountain Car のイメージ

4. 実験設定

実験では，事前学習の効果を分析するため，いくつかのタスクに先の事前学習法を適用し，比較する．タスクの難易度に関わらずデータが成功事例を多く含む場合，事前学習の有無は探索に多くの試行が必要なタスクの方がそうでないタスクと比べて，効果が顕著である可能性が高い．逆に，簡単なタスクでは探索に多くの試行を要さないため，差が表れにくい可能性がある．

タスクに応じて差の表れやすさに違いがあることが推察されるものの，本研究では，簡単のため，比較的学習が容易なタスクを採用する．具体的には，OpenAI gym[Brockman 16] で提供される Pendulum と Mountain Car Continuous（以下，Mountain Car と表記）の 2 つのタスクを採用する．タスクの様子を図 3, 4 に示す．Pendulumn は倒立振り子を立たせるタスクで，Mountain Car は車に山を登らせるタスクである．

実験では，まず，事前学習を行わない通常の NAF で収束が確認されるハイパーパラメタを特定する．次に，そのハイパーパラメタ下で，事前学習方法を適用させ，学習過程を比較する．事前学習で用いるレコード数は (10000, 30000, 100000) の 3 通りを試す．なお，1 レコードは 1 タイムステップごとに得られるログデータ $\mathbf{s}_t, \mathbf{a}_t, r_t, t_t$ を指す．また，各条件で乱数のシード値を変え 10 回実験を行い，平均と標準偏差から分析する．各実験にてエピソード数の最大は 300 とする．

収束が確認されたハイパーパラメタのうち，特に，[Gu 16] にて重要とされるものについて，Pendulum においては，学習率は $1e-3$，ノイズスケールは 0.3，Batch Normalization[Ioffe 15] の代わりに Layer Normalization[Ba 16] を，学習アルゴリズムは ADAM[Kingma 14] を用いる．Mountain Car においては，学習率は $1e-4$ を用い，その他は Pendulum と同様のものを用いる．

一定以上の水準の操作ログデータの準備には，学習済みの方策を用いる．一定以上の水準の操作であることの評価には，エピソード中の累積報酬を用いる．エピソード中の累積報酬に対して閾値を定めて，該当するデータを抽出する．報酬の水準は各タスクによって異なるため，タスクごとに閾値を定める．Pendulum は −200，Mountain Car は 90 を閾値とする．

5. 実験結果

タスクおよび学習方法ごとの累積報酬を表 1 に示す．累積報酬は 300 エピソード目の値について，平均 ± 標準偏差の形式で記載した．

Pendulum では事前学習法 C で，Mountain Car では事前学習法 A, C, D で累積報酬が高かった．逆に，事前学習法 B を適用した場合，事前学習を行わない場合よりも，得られた方策が悪くなる傾向が見られた．この現象は [Gu 16] でも指摘されていた．NAF は，学習に用いるデータに含まれるいい事例と悪い事例から学習するため，単に，いい事例のデータだけからだけでは，いい方策を学習することは難しいと推察される．

Pendulum について，オフライン強化学習を含む事前学習法において，少ないレコード数でも適用しないよりは，適用した方が性能がいい傾向になった．しかし，100000 と多くのログデータを事前学習に用いる場合，その 3 分の 1 程度の量を用いる場合よりも，得られた方策は悪い傾向にあった．このことから，一定以上の水準の操作ログデータを用いて Q 関数全体を事前学習させる場合，学習に用いるデータ数や学習のエポック数を調整することで，得られる方策が改善される可能性があることが推察される．

Mountain Car については，事前学習に用いるレコード数で大きな差は見れられなかった．タスク自体が簡単で，今回設定したレコード数の最小値である 10000 でも十分なデータ量となってしまい，それ以上のレコード数にて差が見られなくなってしまった可能性がある．

次に，各事前学習法について，学習過程を定性的に評価する．エピソードと累積報酬の推移を図 3, 4 に示す．実線が累積報酬の平均を，帯が標準偏差の領域を指す．Pendulum は事前学習に用いるレコード数を 30000 とした学習過程，Mountain Car は事前学習に用いるレコード数を 10000 とした学習過程である．

Pendulum の方では，事前学習法を適用することで，学習初期の累積報酬の値が高いことがわかる．特に，方策関数を教師あり学習させたのち，Q 関数全体をオフラインで強化学習させる事前学習法 C においては，顕著である．ただし，事前学習法 B においては，この傾向は弱い．

Mountain Car の方では，事前学習法 B が，顕著に悪い．その他では，多少の差はあるものの，累積報酬の平均に大きな差はない．ただ，事前学習法 C や D を適用した学習過程では，累積報酬の推移が他のものと比べて，多少安定しているようにも見える．

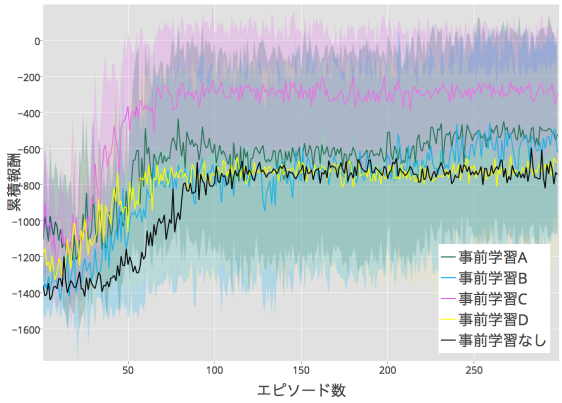

図 3: タスク Pendulum における各学習方法における累積報酬の推移

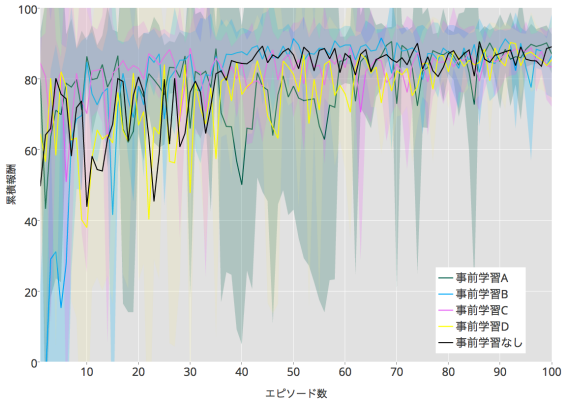

図 4: タスク Mountain Car における各学習方法における累積報酬の推移

2 つのタスクの結果から鑑みるに，事前学習は学習を早め，安定性を向上させる効果を期待でき，有効である可能性が高い．本研究で調査した事前学習方法の中では，方策関数のみを教師あり学習させたのちオフラインで Q 関数全体を強化学習させる，事前学習法 C が最も有用かもしれない．詳細な分析は今後の課題とし，それをもって結論づけたい．

6. 考察と今後の方針

深層強化学習の進展は目覚ましい．本研究では，NAF を用いたが，研究を進める中でも新しい手法が提案されている．例えば，ACER[Wang 16] は，さらに有効である可能性がある．今後は，NAF 以外の手法においても，同様の議論をしていきたいと考えている．

本研究では，実環境への応用を想定して，議論した．しかし，実験では，シミュレータを活用した．最初のステップとしてシミュレータを用いるとしても，実環境への応用を想定したこうした議論や研究は，実環境への応用の成否を踏まえて，行われるべきであると考える．今後は，実環境への応用も含めて，研究を進めていきたい．

謝辞

本研究は JSPS 科研費 JP25700032 の助成を受けたものです．

参考文献

[Ba 16] Ba, J. L., Kiros, J. R., and Hinton, G. E.: Layer normalization, *arXiv preprint arXiv:1607.06450* (2016)

[Brockman 16] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W.: OpenAI gym, *arXiv preprint arXiv:1606.01540* (2016)

[Gu 16] Gu, S., Lillicrap, T., Sutskever, I., and Levine, S.: Continuous deep q-learning with model-based acceleration, *arXiv preprint arXiv:1603.00748* (2016)

[Ioffe 15] Ioffe, S. and Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167* (2015)

[Kingma 14] Kingma, D. and Ba, J.: Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014)

[Levine 16a] Levine, S., Finn, C., Darrell, T., and Abbeel, P.: End-to-end training of deep visuomotor policies, *Journal of Machine Learning Research*, Vol. 17, No. 39, pp. 1–40 (2016)

[Levine 16b] Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D.: Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection, *arXiv preprint arXiv:1603.02199* (2016)

[Lillicrap 15] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D.: Continuous control with deep reinforcement learning, *arXiv preprint arXiv:1509.02971* (2015)

[Mnih 13] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M.: Playing atari with deep reinforcement learning, *arXiv preprint arXiv:1312.5602* (2013)

[Schulman 15] Schulman, J., Levine, S., Abbeel, P., Jordan, M. I., and Moritz, P.: Trust Region Policy Optimization., in *ICML*, pp. 1889–1897 (2015)

[Silver 16] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529, No. 7587, pp. 484–489 (2016)

[Wang 16] Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and Freitas, de N.: Sample Efficient Actor-Critic with Experience Replay, *arXiv preprint arXiv:1611.01224* (2016)