The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

On/off-policy のハイブリッド深層強化学習とシミュレーション環境での制御問題への応用

On/off-policy Hybrid Deep Reinforcement Learning and Simulation in Control Tasks

王 伯楠 *1　河合 新 *1　延原 肇 *1
Bonan Wang　Shin Kawai　Hajime Nobuhara

*1筑波大学大学院システム情報工学研究科知能機能システム専攻
Intelligent Interaction Technologies, Graduate School of Systems and Information Engineering, University of Tsukuba

Recently, deep reinforcement learning with neural network shows great performance in tasks such as game AI and robotics control tasks. However, on-policy and off-policy reinforcement learning methods proposed in related works have problems such as slow exploration speed. To solve these problems, we propose a hybrid deep reinforcement learning method which combines on-policy and off-policy reinforcement learning in this paper. The comparison experiment shows that the proposed method outperforms classic DDPG and DPPO method with an obvious advantage.

1. はじめに

近年、ニューラルネットワークを用いた深層強化学習は様々な分野で成果を出している [Lillicrap 15]。特にゲーム AI や Robotics コントロール問題では素晴らしい性能を示している [OpenAI 18b]。ニューラルネットワークは高い汎用性を持つ近似器として幅広い、かつ、複雑な問題に適用される。

強化学習アルゴリズムは主に on-policy と off-policy の二種類に分けられる。On-policy のアルゴリズムでは探索するエージェントと訓練するエージェントが同一である一方、off-policy のアルゴリズムではそれらは異なっている。On-policy のアルゴリズムは off-policy のアルゴリズムと比べて訓練が速く、安定性も高い。一方で、on-policy アルゴリズムは訓練が進むと、学習と探索の速度が遅くなる。Off-policy のアルゴリズムは過去の経験から学習するため、この問題を緩和できる。しかし off-policy のアルゴリズムは on-policy のアルゴリズムと比べて安定性が悪い。

本稿では、従来の on-policy と off-policy の深層強化学習の問題を解消するため、1) ハイブリッドなエージェントと 2) 訓練アルゴリズムを提案する。具体的には、1) ハイブリッドなエージェントとして、汎用性と効率的な学習のため、Actor-Critic 型エージェントを採用する。さらに、2) その訓練アルゴリズムとして、長期経験と短期経験の両方を活用するため、データを複数の短い時系列になるよう分割して、エージェントを学習させる。また、この時系列での訓練アルゴリズムに対応するために、ネットワークとして LSTM(Long Short-Term Memory) を用いる。提案手法では、目的関数に DDPG の価値関数と DPPO(Distribute Proximal Policy Optimization)[Schulman 17] の価値関数の両方を用いる。

本稿は次のように構成される。第 2 章では関連する従来研究を紹介する。第 3 章では提案手法を説明する。第 4 章では評価実験をその結果を示す。最後に第 5 章で結論を述べる。

2. 関連研究

この章では、主に関連研究の DDPG と DPPO アルゴリズムを説明する。

連絡先: 王 伯楠，筑波大学大学院システム情報工学研究科知能機能システム専攻，wang@cmu.iit.tsukuba.ac.jp

2.1 DDPG(Deep Deterministic Policy Gradient)

DDPG アルゴリズム [Lillicrap 15] は典型的な off-policy の Actor-Critic アルゴリズムである。DDPG エージェントは行動を決めるネットワーク π と評価ネットワーク Q で構成されている。π は状態の観測値 $s \in \mathbb{R}^m$ を入力とし、行動 $a \in \mathbb{R}^n$ あるいは行動の分布を出力する。m, n はそれぞれ状態空間と行動空間の次元数である。Q ネットワークは状態 s と対応する行動 a を評価し、評価値 $q \in \mathbb{R}$ を出力する。

訓練する時、まず Q を更新する。Q を訓練した後、それを教師とし、π を訓練する。π は q を最大化するように θ^π を更新する。DDPG は Replay Buffer から経験データをランダムでサンプリングして訓練する。よって、DDPG アルゴリズムは自由に探索できるが、動作の変化が激しかったり、収束が遅かったりするなどの問題がある。

2.2 DPPO(Distribute Proximal Policy Optimization)

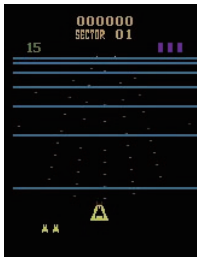
DPPO アルゴリズム [Schulman 17] は典型的な on-policy のアルゴリズムである。DPPO エージェントは行動ネットワーク π と価値ネットワーク V で構成されている。π ネットワークは状態の観測値 s を入力とし、行動の確率分布 N^n を出力する。V ネットワークは状態の観測値ベクトル s を入力とし、価値の予測値 $v \in \mathbb{R}$ を出力する。

DPPO アルゴリズムは最近の経験データしか使えない。よって、DPPO アルゴリズムは常に新しいデータが必要である。DPPO アルゴリズムは学習がはやく、かつ安定性も高い。しかし、DPPO は訓練が進むと探索が遅くなる。

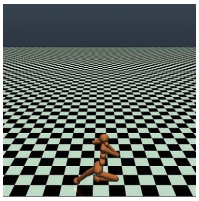
3. 提案手法

3.1 ネットワーク構造

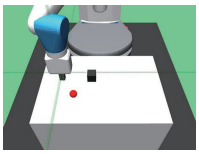
従来法の問題を解消するため、本論文は短期経験と長期経験の両方を活用した on/off-policy のハイブリッドエージェントと強化学習アルゴリズムを提案する。提案手法では Actor-Critic 型エージェントを採用する。短期と長期経験の特徴に対応して、行動策略部分の Actor は LSTM ネットワークを採用する。π は一連の状態の観測値 $s_t, s_{t-1}, \ldots, s_{t-K+1}$、LSTM 初期細胞状態 c_{init} と初期出力 h_{init} を入力とし、行動の確率分布を出力する。ただし $t \in \{1, 2, \ldots, T\}$ は時刻あるいはステップ



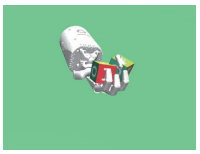
(a) Atari



(b) Humanoid



(c) FetchPush



(d) HandManipulateBlock

図 1: Environments Example

数を表し、下添え字 t はステップ t における値を表すこととする。また、T は最後のステップを表す。$k \in \{1, 2, \ldots, K\}$ は LSTM セルの順番数を表し、K は LSTM のセル数を表す。

$$\pi_t(s_t, s_{t-1}, \ldots, s_{t-K+1}, c_{init}, h_{init}) \sim N^n. \tag{1}$$

LSTM の各ゲートの活性化関数の前に Layer Normalization を用いる。各セルの出力に全結合層をつけて最後の出力にする。

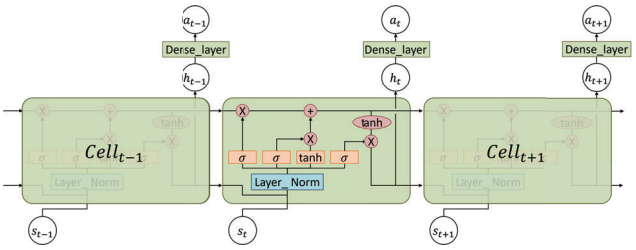


図 2: π ネットワーク構造

価値関数部分の Critic ネットワークは MLP を採用し、Q とする。DDPG の Critic ネットワークと同じく s と a を入力とし、価値の予測値を出力する。負数の報酬値に対応するため、活性化関数は Leaky Relu を用いる。

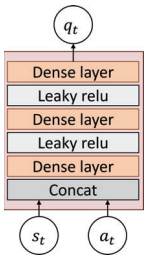


図 3: Q ネットワーク構造

安定性のため、ネットワークは main ネットワークと target ネットワークで構成されている。main ネットワークは学習によって更新され、target ネットワークは main ネットワークのパラメーターを用いて更新される。

3.2 目的関数

長期経験と短期経験の両方を活用するため、それぞれ Q と π の目的関数を設計した。短期経験を使う on-policy 的な目的関数は $L^{vf}(\theta^Q)$ と $L^{pg}(\theta^\pi)$ があり、長期経験を使う off-policy な目的関数は $L^Q(\theta^Q)$ と $L^\pi(\theta^\pi)$ がある。ただし、θ^π, θ^Q はそれぞれのネットワークの重みなどのパラメーターを表している。アルゴリズムはこの 4 つの目的関数を最適化する。

on-policy の価値目的関数 $L^{vf}(\theta^Q)$：

$$L^{vf}(\theta^Q) = \mathbb{E}(Q(s_t, a_t) - Q_t^{'})^2 \tag{2}$$

where (3)

$$Q_t^{'} = \sum_{i=t}^{T}(\gamma^{i-t} r_i) + Q(s_{T+1}, \pi(s_{T+1})). \tag{4}$$

on-policy の行動目的関数 $L^{pg}(\theta^\pi)$：

$$L^{pg}(\theta^\pi) = \mathbb{E}\left(\frac{\pi(a_t \mid s_t)}{\pi^{'}(a_t \mid s_t)} * \hat{A}_t\right), \tag{5}$$

$$\hat{A}_t = \delta_t + \gamma\lambda\delta_{t+1} + \ldots + (\gamma\lambda)^{T-t}\delta_T, \tag{6}$$

$$\delta_t = r_t + \gamma * Q_{t+1} - Q_t. \tag{7}$$

ただし、$\pi^{'}$ は target ネットワークの π を表す。off-policy の価値目的関数 $L^Q(\theta^Q)$：

$$L^Q(\theta^Q) = \mathbb{E}(Q(s_t, a_t) - c_1 * Q_1^{'} - c_2 * Q_2^{'})^2, \tag{8}$$

$$Q_1^{'} = r_t + \gamma * Q(s_{t+1}, \pi(s_{t+1})), \tag{9}$$

$$Q_2^{'} = r_k + \gamma * r_{k+1} + \ldots + \gamma^{K-k} * r_K + \tag{10}$$

$$\gamma^{K-k+1} * Q(s_{K+1}, \pi(s_{K+1})). \tag{11}$$

ただし、c_1 と c_2 は重み引数である。

off-policy の行動目的関数 $L^\pi(\theta^\pi)$：

$$L^\pi(\theta^\pi) = \mathbb{E}(Q(s_t, \pi(s_t))). \tag{12}$$

$L^Q(\theta^Q)$ と $L^{vf}(\theta^Q)$ を最小化し、$L^\pi(\theta^\pi)$ と $L^{pg}(\theta^\pi)$ を最大化する。

3.3 訓練アルゴリズム

訓練する時、先ず複数のエージェントを使って探索し、データを収集する。各エージェントは環境の中で一定のステップ数 T を実行し、これを一つの iteration とする。収集したデータ $(s_t, a_t, r_t, s_{t+1}, c_t, h_t)$ を Replay Buffer $\mathcal{M}$ に保存する。探索が効率よく進むため、各エージェントは探索の途中、小さい確率 ϵ でランダムな行動を取る。ϵ はエージェントによって違う。安定のため、target ネットワークを用いて探索を進め、main ネットワークを更新する。

4. 評価実験

4.1 実験条件

評価実験は深層強化学習においてベンチマーク的なタスク Humanoid タスクを用いる。Humanoid タスクの目的はシミュ

Algorithm 1 on/off-policy algorithm

initialize Replay Buffer $\mathcal{M}$;
initialize Network $\pi_{main}, \pi_{target}, Q_{main}, Q_{target}$;
for i **in** $Max\_Iterations$ **do**
  for w **in** $N\_Workers$ **do**
    reset Iteration Buffer $\mathcal{I}$;
    for t **in** T **do**
      $a_t, c_t, h_t \leftarrow \pi_{target}(s_t)$;
      $a_t \leftarrow Random\ Action\ with\ \epsilon$;
      $s_{t+1}, r_t \leftarrow Environment(a_t)$;
      store $(s_t, a_t, r_t, s_{t+1}, c_t, h_t)$ in $\mathcal{I}$;
    end for
    store $\mathcal{I}$ in $\mathcal{M}$;
    update θ^Q_{main} with $L^{vf}(\theta^Q)$;
    update θ^π_{main} with $L^{pg}(\theta^\pi)$;
    $\theta^Q_{target} \leftarrow (1-\tau) * \theta^Q_{target} + \tau * \theta^Q_{main}$;
    $\theta^\pi_{target} \leftarrow (1-\tau) * \theta^\pi_{target} + \tau * \theta^\pi_{main}$;
  end for
  sample random transition $Batch$;
  update θ^Q_{main} with $L^Q(\theta^Q)$;
  update θ^π_{main} with $L^\pi(\theta^\pi)$;
  $\theta^Q_{target} \leftarrow (1-\tau) * \theta^Q_{target} + \tau * \theta^Q_{main}$;
  $\theta^\pi_{target} \leftarrow (1-\tau) * \theta^\pi_{target} + \tau * \theta^\pi_{main}$;
end for

レーション環境の中の人間を前進することを学習させることである。エージェントは人間の各関節の動作を決める。

表 1: 実験条件

Deep Learning Library	Tensorflow 1.9.0
Simulation Library	openAI gym
Env	Humanoid-v2

訓練する時、Worker ことに on-policy の目的関数を用いてエージェントを更新する、五つの Worker ことに off-policy の目的関数を用いてエージェントを更新する。

表 2: Hyper Parameters

T	1024
N_Worker	5
π learning rate	0.00004-0.000001
Q learning rate	0.00008-0.000002
batch_size	32
epoch	5
sample_size	640
τ	0.2

4.2 実験結果

本研究はベンチマークタスク Humanoid 環境を用いて、従来の DDPG、DPPO と提案手法に対して比較実験を行いました。実験の結果は図 4 に示す。DDPG アルゴリズムはランダムな短い経験だけ用いるため、長い時系列の中の行動の関連性を無視している。よって、エージェントは最初から行動が固定し、前に進まない。DPPO アルゴリズムは学習ができるが、学習の速度が遅い。

提案手法では、安定かつ高速な学習を実現している。提案手法の on-policy 部分は安定な学習を実現し、off-policy 部分はその上探索を進ませ、学習を加速させたと思われる。

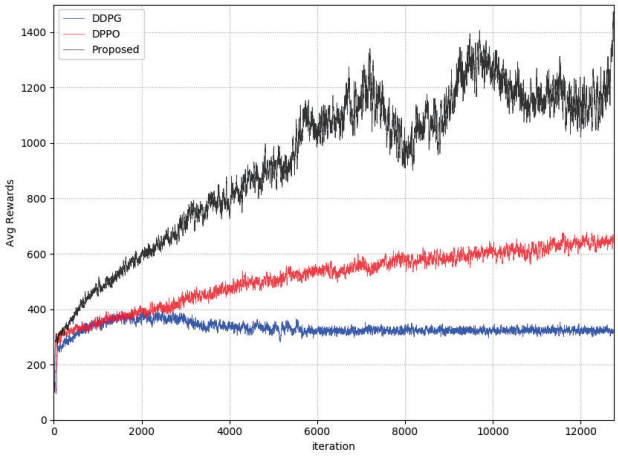


図 4: **Performance Comparison** of DDPG, DPPO and Proposed Method on benchmark task **Humanoid**

5. おわりに

本研究は従来の深層強化学習アルゴリズムの問題点を解決するため、短期経験と長期経験の両方を活用する on/off-policy のハイブリッド深層強化学習アルゴリズムを提案した。Humanoid タスクにおいて、提案手法は従来手法より優れた性能を示した。

Humanoid タスクのような、状態空間と行動空間は連続で、報酬値も丁寧に設計されているタスクに対して、提案手法は有効であると思われる。しかし現実の問題おいて、報酬値は二進値のタスクは多数存在している。エージェントの汎用性を向上させるため、これからは HER[Andrychowicz 17] 手法の投入と Robotics タスク [OpenAI 18a] への実装を検討している。

参考文献

[Andrychowicz 17] Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W.: Hindsight Experience Replay, *CoRR*, Vol. abs/1707.01495, (2017)

[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, *CoRR*, Vol. abs/1509.02971, (2015)

[OpenAI 18a] OpenAI, : Ingredients for Robotics Research (2018)

[OpenAI 18b] OpenAI, : Learning Dexterous In-Hand Manipulation, *CoRR*, Vol. abs/1808.00177, (2018)

[Schulman 17] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O.: Proximal Policy Optimization Algorithms, *CoRR*, Vol. abs/1707.06347, (2017)