2B1-4

モデルベース学習を活用したDDPGのサンプル効率分析

An Analysis of sample efficiency of DDPG with model-based learning

塩谷碩彬 *1 Hiroaki Shioya　那須野薫 *1 Kaoru Nasuno　松尾豊 *1 Yutaka Matsuo

*1東京大学
The University of Tokyo

In this paper, we incorporate model-based learning into Deep Deterministic Policy Gradient that is off-policy and model-free actor-critic method to reduce sample complexity. Experimental result indicates that it is effective to generate synthetic on-policy samples under the learned model in the early stage of learning.

1. はじめに

深層強化学習は近年様々な領域で大きな進歩をとげている. ゲーム [Mnih 2015] や, シミュレーション上での運動 [Lillcrap 2016], 現実世界でのロボット操作 [Levine 2016] といった領域で深層強化学習を用いた成果が報告されている. 深層強化学習を用いることによって人間に匹敵する性能をもつ方策を学習することが可能となる事例も報告されている.[Mnih 2015, Silver 2016]

しかし, 深層強化学習を現実世界のタスクに用いる際にはいくつかの課題がある. 一つは, 学習に必要なコストが大きいという点である. 一般的に, 深層強化学習には多数の試行回数が必要となる. 現実世界で多数の試行を行うのは, 労力や時間を多大に要することとなる. また, 安全性の問題もある. 深層強化学習を行う際には, 事前に何らかの情報がない限り, 学習初期の行動はランダムに近いものとなる. 例えば自動車ような大型の機械をランダムな操作で動かすのは危険が伴うため, こういった領域に深層強化学習を適用する際には障害となる. これらの内, 試行回数の問題は, 操作する対象に関わらず幅広いタスクにおいて深層強化学習を適用する際に問題となり得る.

深層強化学習に必要な試行回数を減らすという観点では, モデルベースの強化学習アルゴリズムとモデルフリーの強化学習アリゴリズムという分類を考えることは有用である. モデルベースの強化学習アルゴリズムは, 現在の状態と行動から次の状態と得られる報酬を予測する環境のモデルをもち, そのモデルに基づいて状態価値を算出する. これに対してモデルフリーの強化学習アルゴリズムは, 環境のモデルを持たず試行回数の結果によって状態価値を算出する. 環境のモデルを活用することで, モデルベースのアルゴリズムは, モデルフリーのアルゴリズムと比較して, 学習に必要な試行錯誤が少なくて済む傾向にある. 一方で, モデルベースのアルゴリズムは, 環境のモデルの性能によって, 獲得できる方策の性能が限定されてしまうという欠点を持つ.

モデルフリーの強化学習アルゴリズムの利点を残しながら学習に必要な試行回数を少なくする試みの一つとして, モデルベースの強化学習アルゴリズムを組み合わせる方法が提案されている.[Gu 2016] は学習した環境のモデルを使用して on policy なサンプルを生成することで, Normarized Advantage Function(NAF) を用いた強化学習において学習に必要な試行回数を減らすことができることを報告した. しかし, この研究において提案された手法の有効性の検証は NAF に対してのみにとどまっている.

NAF と同じくモデルフリーの深層強化学習アルゴリズムの一つに Deep Deterministic Policy Gradient（DDPG）がある.NAF が Q 学習を用いた手法であるのに対し, DDPG はアクター・クリティックを用いた手法である.DDPG はゲームや, シミュレーション上でのロボットの操作など幅広いタスクに適用可能であることが報告されている.[Lillcrap 2016] また, DDPG は NAF と比較して, 行動空間が高次元であるより複雑なタスクに対してより少ない試行回数で高い報酬を得る方策を獲得できることが報告されている. [Gu 2016]

本研究では, モデルフリーの強化学習における試行回数を減らすために環境のモデルを活用する方法を,Deep Deterministic Policy Gradient(DDPG) に対して適用する. 単振り子の振り上げ問題に適用した実験の結果, DDPG に対しては環境のモデルを用いて on policy なサンプルを生成する Imagination Rollouts が, 特に学習の初期段階においてより少ない試行回数でより高い報酬を獲得する方策を学習できる可能性が示唆された.

2. 関連研究

ここでは, Deep Deterministic Policy Gradient（DDPG), Imagination Rollouts, モデルベース学習による探索, について説明する.

2.1 Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient（DDPG）[Lillcrap 2016] は, モデルフリーの深層強化学習アルゴリズムである. 方策を表すアクターと, 価値関数を表すクリティックは共にニューラルネットワークで表現されている. 強化学習のエージェントのタイムステップ t での状態を s_t, 行動を a_t, 獲得する報酬を r_t, 状態行動価値関数を $Q(s,a)$ とすると, クリティックのパラメータは下記の損失関数 L を最小化するように更新される.

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$$

$$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

また, アクターのパラメータは下記の勾配を用いて更新される.

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

連絡先: 塩谷碩彬, 東京大学,shioya@weblab.t.u-tokyo.ac.jp

ここで θ^Q, θ^μ はそれぞれクリティックネットワーク, アクターネットワークのパラメータであり, $\theta^{Q'}$, $\theta^{\mu'}$ はそれぞれ, クリティックのターゲットネットワーク, アクターのターゲットネットワークのパラメータ, γ は割引率を表す.

2.2 Imagination Rollouts

Imagination Rollouts[Gu 2016] は, 環境のモデルを用いて学習に使用するサンプルを生成する手法である.[Gu 2016] はこの手法を用いて, 現実世界での off policy による探索に加えて on policy なサンプルを生成することによって, NAF による学習がより少ない試行回数で進むことを報告した.Imagination Rollouts に用いる環境のモデルはいくつか考えられる. 例えば, ニューラルネットワークや, 時刻 t 毎に変化する線形近似を用いたモデル化が考えられる.[Gu 2016] においては, Imagination Rollouts に用いる環境のモデルにはニューラルネットワークよりも, 時刻 t 毎に変化する線形近似を用いたモデルが有効であることを報告している.

2.3 モデルベース学習による探索

[Gu 2016] はモデルベース学習で得られた方策を探索に活用している. ある確率で, モデルフリー学習で得られた方策に従う代わりに, モデルベース学習によって得られた方策に従った行動を行い, 得られたサンプルをモデルフリーの方策の学習に用いる. どちらの行動に従うかは, 各エピソードの初めに選択される. [Gu 2016] は, NAF においては, この手法で探索を行った場合, 学習に必要な試行回数や得られる報酬は, わずかしか改善しなかったことを報告している. [Gu 2016] では, 探索に用いるモデルベースの学習アルゴリズムとして, [Levine 2014] で用いられていた iLQG を用いた.

3. アルゴリズム

本研究で有効性を検証する, DDPG に対して [Gu 2016] で提示されたモデルベースの学習方法を組み合わせるアルゴリズムを *Algorithm*1 に示す.

4. 実験設定

単振り子の振り上げ問題に提案手法を適用した. シミュレータ環境には OpenAI gym[Brockman 2016] の Pendulum-v0 を使用した. タスクのイメージを図 1 に示す. ただし, 今回の実験にあたって設定を 2 カ所変更した.1) 環境から受け取る状態を変更した. 元の設定では, 振り子の状態を $cos\theta$, $sin\theta$, $\dot{\theta}$ の 3 次元を入力としていたが, 本研究では θ, $\dot{\theta}$ の 2 次元を入力として用いる.2) 振り子に作用できるトルクの範囲を大きくした. 元の設定では [-2, 2] の間になるように設定されているが, この実験では [-10, 10] の範囲で操作できるように設定した. これは iLQG によって得られる方策が, 元の設定では, トルクの制限によって実現されず, 方策の正しい改善がなされなくなってしまうためである.

その他の設定は OpenAI のデフォルトの設定にしたがう.1 エピソードのタイムステップは 200 に設定し, 50 エピソードまで学習を行った.

DDPG のアクターネットワークとクリティックネットワークは共に 2 層のネットワークを用いた. ユニット数は 1 層目が 400, 2 層目が 300 とした. バッチ正則化 [Ioffe 2015] は用いない. 活性化関数には ReLU を用い, 目的関数の最適化には ADAM[Kingma 2014] を用いた. 探索用ノイズは, Ornstein-Uhlenbeck 過程 [Uhlenbeck 1930] によって生成する.

Algorithm 1 DDPG + Imagination rollouts + iLQG による探索

Randomly initialize critic network $Q(s, a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$with θ^Q and θ^μ
Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer $R \leftarrow \emptyset$ and fictional buffer $R_f \leftarrow \emptyset$
additional buffers $B \leftarrow \emptyset$ and $B_{old} \leftarrow \emptyset$ with size nT
Initialize fitted dynamics model $\mathcal{M} \leftarrow \emptyset$
for episode = 1, M **do**
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state s_1
  Select $\mu'(s,t)$ from $\mu(s|\theta^\mu), \pi_t^{iLQG}(a_t|s_t)$ with probabilities $\{p, 1-p\}$
  for t =1, T **do**
    Select action $a_t = \mu'(s,t) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action a_t and observe reward r_t an observe new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in R and B
    if $mod(episode \cdot T + t, m) = 0$ and $\mathcal{M} \neq \emptyset$ **then**
      Sample m(s_i, a_i, r_i, s_{i+1}) from B_{old}
      Use $\mathcal{M}$ to simulate l steps from each sample
      Store all fictional transitions in R_f
    end if
    Sample a random minibatch of m trainsitions (s_i, a_i, r_i, s_{i+1}) I $\cdot$ l times from R_f and I times from R
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss:

$$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

    Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

    Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

  end for
  if B is full **then**
    $\mathcal{M} \leftarrow$ FitLocalLinearDynamics(B_f)
    $\pi^{iLQG} \leftarrow iLQG_OneStep(B_f, \mathcal{M})$
    $B_{old} \leftarrow B, B \leftarrow \emptyset$
  end if
end for

DDPG に対して [Gu 2016] で提示されたモデルベースのアルゴリズムを組み合わせる方法がサンプル効率の向上に有効であるか否かを検証するため, 次の 4 つの方法を比較した. 1)DDPG, 2)DDPG + Imagination Rollouts, 3)DDPG + モ

デルベース学習による探索 (DDPG + iLQG), 4) DDPG + Imagination Rollouts + モデルベース学習による探索 (DDPG + Imagination Rollouts + iLQG)

各学習方法は, ランダムに選ばれた異なる 3 個の初期状態から行なった結果得られた報酬を平均したもので評価する.

図 1: 単振り子の振り上げタスクのイメージ

5. 実験結果

実験結果について述べる. 図 2 にエピソードが進むにつれて獲得される報酬の推移を示す. また, 表 1 に, 各学習方法ごとに最終エピソードで獲得した報酬をまとめる. iLQG を探索に用いた手法に関しては, 括弧の中に, 最後に方策に DDPG を用いたエピソードで獲得した報酬を記載している. 報酬は平均獲得報酬 ± 標準偏差の形で記載している.

DDPG+Imagination Rollouts については, DDPG と比較して, 学習の早い段階ではより高い報酬を獲得できる方策を学習している. このことから, Imagination Rollouts は DDPG に対しても学習を速めるのに有効である可能性が示唆される. 一方で, 今回学習させたエピソードの中では, 最終的に得られた報酬は DDPG よりも悪くなっている. これは, 学習している環境のモデルが厳密には正しくないため, 実際に得られたデータから十分有効な方策を学習できる状態まで学習が進んだ段階では, 却って方策の改善を妨げてしまうためと考えられる. この問題点を回避するには, [Gu 2016] でも述べられていたように, ある程度 DDPG の学習が進んだ段階で Imagination Rollouts を打ち切ることが一つの方法として挙げられるだろう.

DDPG+iLQG についても, DDPG と比較して, 学習の早い段階ではより高い報酬を獲得できる方策が得られているが, 最終的に得られた報酬は DDPG よりも悪くなっている. この方法が学習の早い段階で良い報酬を獲得できているのは, DDPG の学習より先に, iLQG が良い方策を獲得できているためと推察される. 一方で, 学習の後半には, DDPG が iLQG よりも良い方策を獲得するために, 獲得できる報酬が DDPG と比較して悪くなっているものと思われる. ただし, 先も述べたように, 最後に方策に DDPG を用いたエピソードで獲得した報酬についても, DDPG だけで学習した場合と比較して改善していないため, この実験からでは, iLQG による探索が DDPG によって得られる方策をより改善するのに有効であるとは言えない.

DDPG+iLQG+Imagination Rollouts については, 2 つの方法を組み合わせたことの有効性はそれほど明確ではない. 学習の早い段階では, DDPG + Imagination Rollouts と比較して, 獲得できる報酬はあまり変わらない. また, 学習の進んだ段階では, DDPG + iLQG と比較して獲得できる報酬はあまり変わらず, DDPG + Imagination Rollouts と比較して悪くなっている. ただし, 最後に方策に DDPG を用いたエピソードで獲得した報酬で比較すると, わずかに多い報酬を獲得できている.

これらの結果から, 学習に要する試行回数を少なくするという観点から見れば, DDPG + Imagination Rollouts は有効である可能性が示唆される. 一方で, iLQG による探索は, 学習の早い段階では DDPG より良い方策で行動することができるものの, 最終的に獲得できる報酬は悪くなってしまう可能性が示唆される. いずれの場合も, 最終的に得られる報酬を悪くしないために, モデルベースの学習をどの時点まで活用するかが重要な課題であると考える.

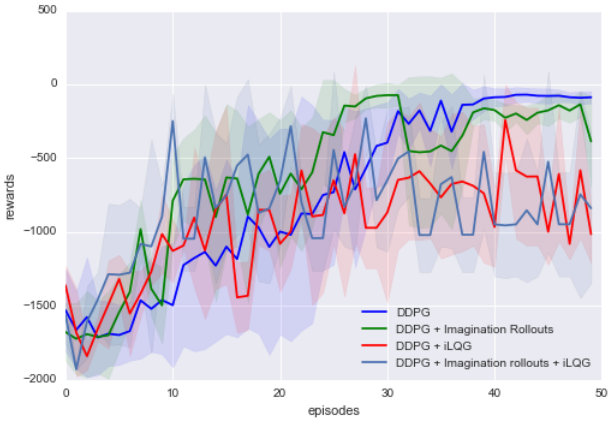

図 2: 各学習方法における獲得報酬の推移

表 1: 単振り子振り上げタスクの学習方法ごとの結果比較

手法	報酬
DDPG	-87 ± 41
DDPG + Imagination Rollouts	-385 ± 310
DDPG + iLQG	$-1014 \pm 199(-643 \pm 759)$
DDPG + Imagination Rollouts + iLQG	$-839 \pm 506(-263 \pm 332)$

6. まとめ

本研究では,Imagination Rollouts を DDPG に対して組み合わせた手法の有効性の検証を試みた. 単振り子の振り上げタスクに対して, DDPG に対してはモデルベースの強化学習によって獲得された方策を探索に使用する方法と, 環境のモデルを用いて on policy なサンプルを生成する Imagination Rollouts を DDPG に組み合わせた学習方法を適用した結果, 特に Imagination Rollouts が学習の初期において学習を速める可能性が示唆された. 今後は, 獲得できる報酬を減らさず学習に要する試行回数が減らせるようなモデルベース学習の活用方法の研究や, より多くのタスクに対して本研究で検証したアルゴリズムを適用し, DDPG をモデルベース学習と組み合わせる学習方法の有効性が同様に確認されるか否かの検証を行っていく予定である.

謝辞

本研究は JSPS 科研費 JP25700032,JP16H06562 の助成を受けたものです.

参考文献

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

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

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

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

[Levine 2014] Levine, S., & Abbeel, P. (2014). Learning neural network policies with guided policy search under unknown dynamics. In Advances in Neural Information Processing Systems (pp. 1071-1079).

[Levine 2016] Levine, S., Finn, C., Darrell, T., & Abbeel, P. (2016). End-to-end training of deep visuomotor policies. Journal of Machine Learning Research, 17(39), 1-40.

[Lillcrap 2016] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971. [Mnih 2015] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.

[Silver 2016] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... & Dieleman, S. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484-489.

[Uhlenbeck 1930] hlenbeck, G. E., & Ornstein, L. S. (1930). On the theory of the Brownian motion. Physical review, 36(5), 823.