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

進化的計算と方策勾配法による学習を用いた 3次元制御タスクにおけるマルチタスク深層強化学習

Multi-task Deep Reinforcement Learning with Evolutionary Algorithm and Policy Gradient Method in 3D Control Tasks

今井 翔太 Shota IMAI　清 雄一 Yuichi SEI　田原 康之 Yasuyuki TAHARA　大須賀 昭彦 Akihiko OHSUGA

電気通信大学大学院情報理工学研究科
Graduate School of Informatics and Engineering. The University of Electro-Communications

In deep rerinforcement learning, it is difficult to converge when the exploration is insufficient or a reward is sparce. Besides, in a specific tasks, the number of exploration may be limited. Therefore, it is considered effective to learn in source tasks previously to promote learning in the target tasks. In this research, we propose a method to train a model that can work well on variety of target tasks with evolutionary algorithm and policy gradient method. In this method, agents explore multiple environments with diverce set of neural networks to train a general model with evolutionary algorithm and policy gradient methid. In the experiments, we assume multiple 3D control source tasks. After the model training with our method in the source tasks, we shows how effective the model is for the 3D Control tasks of the target tasks.

1. はじめに

近年，深層学習と強化学習を組み合わせた深層強化学習がゲーム AI[Mnih 15] や，実世界における機械制御 [Levine 16] 等の分野で大きな成果をあげている．深層強化学習は，深層ニューラルネットワークを関数近似器として用い，Q 値や方策の出力を学習する手法である．

深層強化学習においては，学習のために膨大な探索によるサンプルを必要とする．しかし，探索する環境の行動・状態空間が大きすぎる場合は，十分なサンプルを得るのに大きな時間がかかり，得られたサンプルによっては収束しないこともある．実世界において実機を用いた探索を行う場合には，物理的な条件による制約から，複数の探索自体が困難である．また，学習が完了していない方策で探索を行うことで，機器が危険な行動を行う可能性がある．この問題を解決するため，解きたいタスク（ターゲットタスク）とは別のタスク（ソースタスク）で事前に学習を行うことでニューラルネットワークの転移可能なパラメータを獲得し，解きたいタスクにおいて少数のサンプルで高速に学習が可能な汎用的学習モデルを作成することが望ましい．

深層強化学習で学習を行いたいタスク（ターゲットタスク）があるとき，最終的に解きたいタスクと性質が似ており，なんらかの理由（単にタスクが単純，シミュレータで学習可能）で学習が容易なタスク（ソースタスク）が他に存在するのであれば，ソースタスクで学習を行うことで，効率的に両タスクに共通するパラメータを獲得できる可能性がある．また，ソースタスクが複数ある場合，それら全てのタスクに対して高いパフォーマンスを発揮するようなパラメータの学習を行うことで，ソースタスク，ターゲットタスクの広い範囲のタスクで共通する良いパラメータを学習できると考えられる．

従来の研究では，勾配降下を用いた学習を工夫するなどして，複数のタスクで高速な学習が可能なパラメータを得る事前学習手法が提案されてきた [Finn 17]．しかし，事前学習を行うタスクで報酬がスパースである場合の学習の難しさ [Sutton 98]，強化学習の一般的な課題であるサンプルの偏り [Henderson 17]，初期パラメータへの依存 [Kolen 90] に対応した事前学習手法の検討は不十分である．

これらの背景を踏まえ，本研究では，進化的計算のランダム要素と，勾配降下による学習を組み合わせ，上記の課題を解決しつつ事前学習可能な手法を提案する．本研究では，解くべきタスクを 3 次元制御問題と定め，行動空間が連続である場合に用いられる深層強化学習アルゴリズムである Deep Deterministic Policy Gradient(DDPG)[Lillicrap 15] を用い，ニューラルネットワークに事前学習手法を適用する．

連絡先: 今井翔太，電気通信大学大学院情報理工学研究科，東京都調布市調布ヶ丘 1-5-1, Email:imai.shota@ohsuga.lab.uec.ac.jp

2. 関連研究

本研究では，行動空間が連続な場合に使用される深層強化学習アルゴリズムとして，DDPG[Lillicrap 15] を用いる．DDPG は，状態の入力に対して，決定論的に行動を出力する方策をパラメータで表し，期待報酬和が最大になるような勾配を用いてパラメータを更新する深層強化学習アルゴリズムである．行動空間が連続な場合における Q 学習 [Watkins 92] では，一般的に特定の状態において最も Q 値が高い行動を求めることが困難である．一方 DDPG は入力に対して，それぞれの行動の出力で，決定論的に一つの値を出力するため，主に行動空間が連続である場合のタスクで使用される．DDPG のアーキテクチャは，行動の数と同数の出力を持ち，観測した状態を入力とすることで行動の値を出力する Actor と，観測した状態と Actor の出力を入力とし，観測した状態における Actor の出力を評価する Critic で構成される．Critic は一般的な教師あり学習によって学習を行い，Actor は Critic の出力を使用し，決定論的な方策を学習する方策勾配定理 Deterministic Policy Gradient(DPG)[Silver 14] によってパラメータの更新を行う．パラメータ θ^Q を持つ Critic Q は，リプレイバッファからのサンプル (s_i, a_i, r_i, s_{i+1}) を用いて，以下の損失関数を最小化するようにパラメータを更新する．

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

ここで y_i は，パラメータ $\theta^{Q'}$ を持つ Critic のターゲットネットワーク Q'，パラメータ $\theta^{\pi'}$ を持つ Actor のターゲットネッ

トワーク，割引報酬率 γ を用いて，以下の式で与えられる．

$$y_i = r_i + \gamma Q'(s_{i+1}, \pi'(s_{i+1}|\theta^{\pi'})|\theta^{Q'}) \quad (2)$$

Actor のパラメータは上記の Critic の出力を使用し，以下の式で与えられる勾配によって更新する．

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \Sigma_i \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=a_i} \nabla_{\theta^\pi} \pi(s|\theta^\pi)|_{s=s_i} \quad (3)$$

深層強化学習において複数タスクに適用可能な事前学習手法としては MAML(Model-Agnostic Meta-Learning)[Finn 17] がある．これは，画像認識や強化学習などの深層学習のタスクにおいて，汎用的に使用できる学習モデルのパラメータの初期値を勾配降下法による学習で得る手法である．この手法は，学習の仕方そのものを学習するメタラーニングと呼ばれる手法の一種である．

深層強化学習における効果的な学習として，進化的計算と勾配降下法による学習を組み合わせた Evolutionary Reinforcement Learning(ERL)[Khadka 18] が提案されている．ERL では，進化的計算のランダム要素によって学習するニューラルネットワークの個体群の中に，勾配降下で学習するニューラルネットワークを混ぜ，定期的に勾配降下で学習するネットワークのパラメータを個体群の一部にコピーすることで，一つのタスクに対する最適化を効率化する．進化的計算では，複数のニューラルネットを用いて学習を行うため，学習が安定しているものが最適化され，また，パラメータの初期値に対して頑健であるという特性がある．これに加え，複数の個体による探索を行うため，報酬がスパースなタスクであっても報酬の獲得がしやすいという利点がある．ERL は強化学習における課題を進化的計算を用いて解決しつつ，一般的な勾配降下による学習を取り入れることで，タスク特化の学習を効率化したものといえる．

3 次元における制御課題のように，行動，状態空間が膨大である場合は，報酬がスパースであり，事前学習に必要な探索自体が難しいという問題がある [Sutton 98]．また，単一のニューラルネットワークを用いた学習は，ネットワークの初期パラメータによって学習が左右される傾向がある [Kolen 90]．その他，単一のニューラルネットワークを用いた探索では，得られるサンプルに偏りが発生し，局所解に陥るなどして学習がうまくいかない場合がある [Henderson 17]．これらは，一般的な事前学習手法における「転移可能なパラメータを得られるかどうか」という問題設定からは独立した課題である．そのため，仮に転移可能なパラメータを得る手段があったとしても，タスクによっては，これらの課題により事前学習が阻害される可能性がある．これらの課題は上述の ERL のように進化計算的な手法を用いることで解決が可能である．したがって，ERL のように進化計算を行い，ニューラルネットワークの最適化の対象を複数のタスクに設定することにより，課題を克服しつつ様々なタスクで利用可能なパラメータを取得することができると考えられる．本研究では，進化的計算による最適化と，ソースタスクにおける複数タスクに対する勾配降下の学習を事前学習として行うことで，3D 制御タスクに共通するニューラルネットの良いパラメータを獲得し，ターゲットタスクとなる複数のタスクで高速に学習することを目指す．

3. 提案手法

3.1 問題設定

本手法の問題設定は以下の通りである．

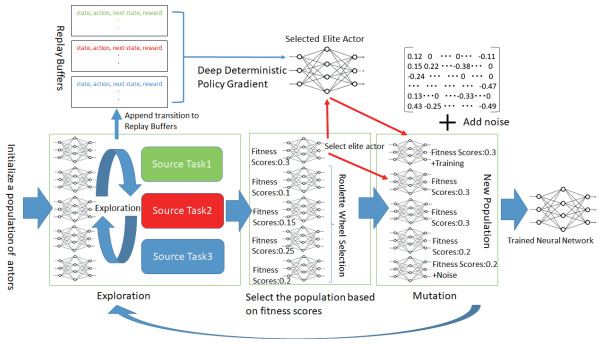

図 1: 進化的計算と方策勾配法を用いた深層強化学習

タスクの質が同じタスクの集合を仮定する．タスクの集合のうち，モデルの事前学習に用いられるものをソースタスクと呼ぶ．タスクの集合のうち，事前学習の時点では与えられず，学習後に解くべき未知のタスクをターゲットタスクと呼ぶ．すなわち，本研究では，ターゲットタスクと似たタスクのソースタスクで事前学習を行うことで，ニューラルネットがタスク間に共通する良いパラメータを獲得し，複数のターゲットタスクにおいて，高速に学習可能なニューラルネットワークを得ることを目指す．

3.2 提案手法の詳細

3.2.1 個体集団による探索

図 1 に提案手法の概要を示す．最初に，進化的計算の個体集合となる複数のニューラルネットワークを用意する．本研究では，状態を入力とし，行動を出力するニューラルネットワーク（以後 Actor） をこの個体集団のニューラルネットワークとする．個体集合 pop_π の Actor π はそれぞれランダムなパラメータ θ^π で初期化される．ソースタスクの集合を T とし，各 Actor π は各世代で全てのタスクを探索し得られた報酬 r_π を記録する．また，この探索で得られたサンプル (s_i, a_i, r_i, s_{i+1}) は各タスクのリプレイバッファに記録する．複数の Actor の探索により，偏りのないサンプルを得ることが期待でき，これらのサンプルは後の DDPG による Actor の学習に使用する．

3.2.2 適応度によるエリート選択，学習

全ての Actor π による探索終了後，記録された報酬の合計を元に各 Actor π の適応度 f_π を計算し，適応度が最も高い Actor をエリート個体として選択する．ここで選択されたエリート個体は，後述のノイズ付加をパスする．選択されたエリート個体の Actor のコピーは，勾配降下による学習で追加の最適化を行うため，確率的に選ばれたタスクのリプレイバッファ R のサンプルを用いて DDPG による学習を行う．ここで，あるタスクのリプレイバッファ R_{T_i} が選択される確率 $P_{R_{T_i}}$ は，その世代の各 Actor が各ソースタスク T_i で得た報酬 r_{T_i} を元に以下の式で与えられる．

$$P_{R_{T_i}} = \frac{r_{T_i}}{\Sigma_i r_{T_i}} \quad (4)$$

3.2.3 適応度による個体の選択，ノイズの付与と最終選択

エリート個体の選別後，次世代の個体集合となる Actor を，適応度を元にルーレット選択によって選択する．ここで選ばれたネットワークは，確率的にノイズを付与することにより，突然変異を行う．エリート個体，コピーされ勾配降下による学習を行ったエリート個体，エリートを除いて選択された個体が次世代の個体集合 pop'_π となり，現世代 pop_π にコピーされる．

Algorithm 1 提案手法の疑似コード

Initialize actor π and critic Q with weight θ^{π} and θ^{Q}, respectively
Initialize a population of k actors pop_{π}
Initialize replay buffers R
Define a random number generator $r() \in [0, 1)$
for $generation = 1, \infty$ **do**
  for actor $\pi \in pop_{\pi}$ **do**
    for all source tasks T_i **do**
      Explore T_i using θ^{π}
      Append transition to replay buffer R respectively
    end for
  end for
  Select the elite actor π based on fitness score f_{π}
  Select the replay buffer R based on all fitness scores f_{π}
  Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
  Update Q by minimizing the loss
  $L = \frac{1}{N}\Sigma_i(y_i - Q(s_i, a_i|\theta^{Q}))^2$
  Update copied elite actor π using the sampled policy gradient
  $\nabla_{\theta^{\pi}} J \approx \frac{1}{N}\Sigma_i \nabla_a Q(s, a|\theta^{Q})|_{s=s_i, a=a_i} \nabla_{\theta^{\pi}} \pi(s|\theta^{\pi})|_{s=s_i}$
  Select the $(k-2)$ actors based on fitness scores f_{π} and insert selected actors into next generation's population pop'_{π}
  for $(k-2)$ actors $\in pop'_{\pi}$ **do**
    if $r() < mut_{prob}$ **then**
      Add noise to θ^{π}
    end if
  end for
  Insert the elite actor into pop'_{π}
  Insert the copied elite actor pop'_{π}
  $pop_{\pi} \leftarrow pop'_{\pi}$
end for

以降，上記の手順を最終世代まで繰り返し，最終世代で得られたエリート個体の Actor が最終的に得られるニューラルネットワークである．

4. 実験

実験では，物理演算エンジン Pybullet [Coumans 19] で提供される OpenAI Gym[Brockman 16] の 3 次元物体制御タスクを用いて，事前学習手法の有効性の評価を行う．

4.1 実験設定

PyBullet で提供されるタスクの中から，Minitaur，Hopper，Walker をソースタスクとし，HalfCheetah，Ant，Humanoid をターゲットタスクとする（図 2）．各タスクはそれぞれ，物体の部位の角度や回転，速度などを状態とし，関節など，物体の各部位の調節を出力とする．出力を調節することで，物体が適切に行動（立ち上がる，進む）すると報酬が与えられる．ソースタスクにおいて，提案手法を用いた複数世代の探索，選択，ノイズ付与，学習を行い，最終世代におけるエリート個体となる 1 つの Actor を全てのターゲットタスクに用いる学習モデルとする．各世代における Actor の個体数は 10 とし，1 つの環境における最大探索数は 1000 回，ノイズ付与のフェーズでは，エリートを除いた次世代個体にそれぞれ 1/4 の確率でランダムに発生させた平均 0 のガウシアンノイズを付与する．パラメータの過度な変動を防ぐため，このノイズの中で-0.5 以下，0.5 以上の値は 0 として扱う．本実験における事前学習の世代数は 100 とする．Actor として用いるニューラルネットワークの設定は，入力と出力はタスクごとに可変とし，隠れ層は 3 層，各層のノード数は 128 とする．Critic として用いるニューラルネットワークの設定は，状態を入力する側のネットワークをノード数 200 の隠れ層 1 層とし，Actor の入力と，状態を入力した側からの出力を受け取るニューラルネットワークは 300 ノードの隠れ層 1 層とし，出力は 1 である．活性化関数としては ReLU を使用し，最適化手法として Adam[Kingma 14] を用いた．

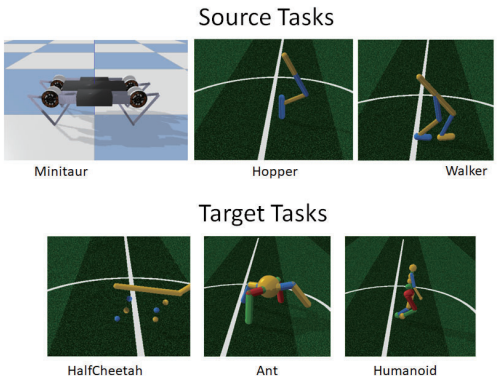

図 2: 実験で用いるソースタスクとターゲットタスク

4.2 実験結果と考察

図 3 に各ターゲットタスクにおける，提案手法によって得られた事前学習モデルと，ランダムに初期化したモデルの学習過程を示す．横軸はエピソード数，縦軸は各エピソードで得られた報酬を指す．Ant，HalfCheetah においては，事前学習モデルを使用した場合のエピソードの初期の段階で獲得する報酬量が，ランダムに初期化したモデルの報酬量を上回っていることが確認できる．両タスク共にエピソードによって得られる獲得報酬量に大きな幅があるものの，全体的に事前学習を行ったモデルの方が大きな報酬を得ており，少ないサンプルで高速に学習できているといえる．一方，Humanoid においては事前学習モデルが，ランダムな初期の場合よりも高いパフォーマンスを発揮できていないことが確認できる．これについては，Humanoid タスクは，他のタスクと異なる多くの部位の観測，行動が要求されるため，ソースタスクでそれらと紐づいた良いパラメータが学習できなかったこと，また今回実験で使用したニューラルネットワークの表現力が足りていないことなどが原因として考えられる．

5. まとめ

本稿では，深層強化学習における既存の事前学習手法において解決されていない課題に対応する手法として，進化的計算と勾配降下を用い，特に 3 次元物体制御タスクにおいて複数のタスクで高速学習可能なパラメータを得る事前学習手法を提案した．実験では，物演算エンジン PyBullet で提供される Open AI Gym の 3 次元制御タスクに提案手法を適用し，複数タスクで高速学習可能なパラメータを持ったニューラルネットが得られることを示した．

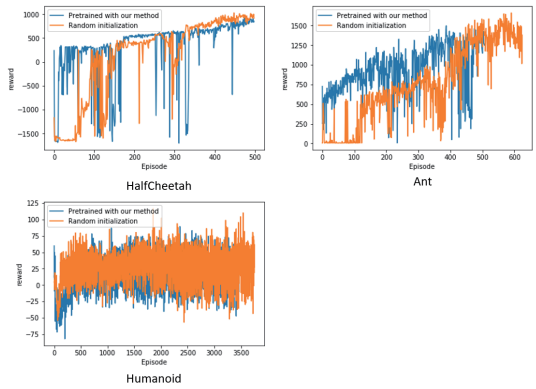

図 3: ターゲットタスクにおける事前学習モデルとランダムに初期化したモデルの学習

今後は，さらに探索空間が大きく，報酬がスパースである課題に対して本手法を適用し，有効性を検証したい．

6. 謝辞

本研究は，JSPS 科研費 16K00419，16K12411，17H04705，18H03229，18H03340 の助成を受けたものです．

参考文献

[Brockman 16] Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W.: OpenAI Gym, *CoRR*, Vol. abs/1606.01540, (2016)

[Coumans 19] Coumans, E. and Bai, Y.: PyBullet, a Python module for physics simulation for games, robotics and machine learning, `http://pybullet.org` (2016–2019)

[Finn 17] Finn, C., Abbeel, P., and Levine, S.: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, in Precup, D. and Teh, Y. W. eds., *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, International Convention Centre, Sydney, Australia (2017), PMLR

[Henderson 17] Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D.: Deep Reinforcement Learning that Matters, *CoRR*, Vol. abs/1709.06560, (2017)

[Khadka 18] Khadka, S. and Tumer, K.: Evolution-Guided Policy Gradient in Reinforcement Learning, in Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. eds., *Advances in Neural Information Processing Systems 31*, pp. 1196–1208, Curran Associates, Inc. (2018)

[Kingma 14] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *CoRR*, Vol. abs/1412.6980, (2014)

[Kolen 90] Kolen, J. F. and Pollack, J. B.: Back Propagation is Sensitive to Initial Conditions, in *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3*, NIPS-3, pp. 860–867, San Francisco, CA, USA (1990), Morgan Kaufmann Publishers Inc.

[Levine 16] 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)

[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)

[Mnih 15] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, No. 7540, pp. 529–533 (2015)

[Silver 14] Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M.: Deterministic Policy Gradient Algorithms, in Xing, E. P. and Jebara, T. eds., *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32 of *Proceedings of Machine Learning Research*, pp. 387–395, Bejing, China (2014), PMLR

[Sutton 98] Sutton, R. S. and Barto, A. G.: *Introduction to Reinforcement Learning*, MIT Press, Cambridge, MA, USA, 1st edition (1998)

[Watkins 92] Watkins, C. J. C. H. and Dayan, P.: Q-learning, *Machine Learning*, Vol. 8, No. 3, pp. 279–292 (1992)