ISSN 2186-5647

P-18

Recurrent Replay Distributed-DQN における サンプル効率の向上

日大生産工　○小菅　敬胤　　日大生産工　山内　ゆかり

1. まえがき

強化学習は産業用ロボット,自動運転システム,ゲームなど幅広い産業で活用されている。

強化学習の代表的な学習手法としてQ学習(Q-Learning)があり、そこに深層学習を取り入れた深層強化学習(Deep-Q-Network:DQN)[1]がある。さらに、本研究の基となる手法にはDQNに分散学習、優先付き経験再生、RNN及びLSTMを導入している。

DQNにはデータ効率を高めることに重要な要素である経験再生がある。しかし、マルコフ決定過程(POMDP)の進歩に伴い、経験再生には改良が必要となった。また、前述した手法にRNNのLSTMを導入し時系列情報を考慮する必要があるが、学習の難しさから結果を残すことができていない。そこで、Steven Kapturowski らは R2D2(Recurent-Replay-Distributed-DQN)[2]として、LSTMを考慮しても学習を安定化することに成功した。しかし、R2D2は強化学習における最新の手法であり、Atari環境において高性能を出力しており、特に問題点が挙げられていない。

本研究では,サンプル効率を向上させるために、優先度の高い経験から学習する仕組みである優先度付き経験再生を改良し、学習させると同時に、進化的手法のエリート保存戦略により、優先度の高い経験を経験メモリ内に保存する手法を提案する。

2. 従来研究

2.1 DQN(Deep Q Network)

DQN には経験再生(Experience Replay)という学習データを経験メモリに保存し、蓄積されたデータからランダムに取り出す手法がある。以下に DQN を表す式(1)(2)を示す。

$$Q(s,a) = R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') \quad (1)$$

$$a' = \ argmax\ Q(s', a') \quad (2)$$

式(1)のs, a, R, s_{t+n}は状態、行動、報酬、次の状態を表している。これらを経験再生に活用する。式(2)は Q 値の最大値を行動選択時に活用することを表している。

2.2 DDQN(Double DQN)

Double DQN[3]はDQNにDouble Q-learningを追加した手法となっている.DQNでは行動選択とQ値の評価をするネットワークが同一であり、Q値が過大評価される傾向にある。そこで行動選択とQ-network、Q値の評価をtarget Q-networkに分割し、低減することができる。以下に式(3)、(4)として示す。

$$Q(s,a) = R_{t+1} + \gamma\ max\, Q_{target}(s', a') \quad (3)$$

$$a' = \ argmax\ Q(s', a') \quad (4)$$

2.3 Prioritized Experience Replay

優先的経験再生[4]は経験メモリに蓄積したデータに優先順位をつけ、重要度の高い経験から一つ、あるいはミニバッチ数分取り出す仕組みとなっている。以下の図1に経験再生と優先付き経験再生を示す。

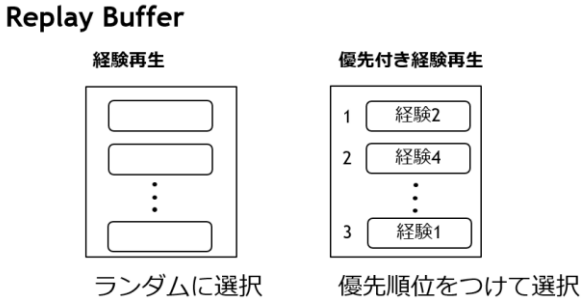

図1 経験再生と優先付き経験再生

2.4 分散型強化学習

分散型強化学習とは、分散学習と強化学習を組み合わせた手法となっている。この学習手法は、Learner(学習者)とActor(行動者)を分離させ、別々の環境で並列処理するプログラムである。多数のActorからデータを収集できる分散学習を取り入れることで性能が向上する。

2.5 Dueling Network

Dueling Network[5]は通常のDQNでは、現在の状態と行動後の状態を出力してQ値を計算していた。この手法では、状態価値関数V(S　)と、ある状態のときに行動するAdvantage関数A(S　,a)に分けてQ値を求める。この計算式(5)として示す。

$$Q(s,a;\theta,\alpha,\beta) = V(s;\theta,\beta) + \left(A(s,a;\theta,\alpha) - \frac{1}{|A|}\sum_{a'} A(s,a';\theta,\alpha) \right) \quad (5)$$

s, aは状態、行動を表している。

Improving sample efficiency in Recurrent Replay Distributed-DQN
Kein KOSUGE and Yukari YAMAUCHI

2.6 Recurrent Neural Network(RNN)

RNNは、時系列データをニューラルネットワークに通して学習させるモデルとなっている。しかしDQNとRNNには経験再生との相性が悪く、エピソード全体の計算コストが膨大になる。そこでLSTMを導入することでこの問題を解決した。LSTMはDueling Networkの畳み込み層の後ろに挿入される。ここでは入力と出力に加え、時系列データを記憶する役割を持つ隠れ状態が存在する。図2にLSTMを導入した概念図を示す。h0、hは初期の隠れ状態、隠れ状態を表している。

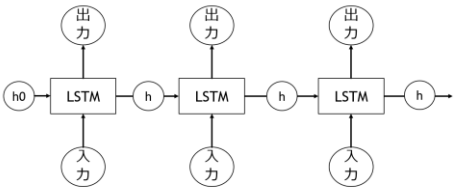

図2　RNNにおけるLSTMの導入

3. 提案手法

本研究では，従来手法に取り入れられている優先度付き経験再生を改良した提案を行う。優先度付き経験再生では優先度の高い経験から学習する仕組みとなっているが、その経験を学習させると同時に、進化的手法のエリート保存戦略により、優先度の高い経験を経験メモリ内に保存する手法を提案する。ある一定領域に達したら、そのメモリから TD 誤差を求め、優先順位をつけて学習する仕組みを設ける。

4. 実験および検討

本研究では、探索問題を考慮し、環境10×10の迷路環境とする。 また、経路に障害物を導入することで、学習する様子が明確になると考える。実験環境を以下の図3に示す。

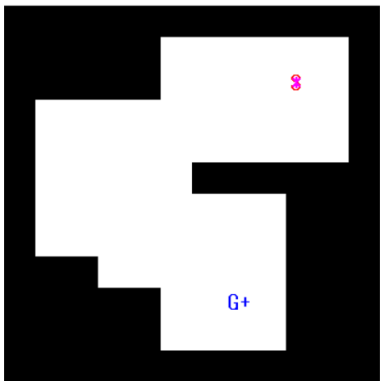

図3 実験環境

Sがスタート、G＋をゴールとする。ゴールに達したら報酬を与える仕組みにする。

5. まとめ

本研究では優先度付き経験再生に進化的手法のエリート保存戦略により、優先度の高い経験を経験メモリ内に保存する手法を提案した。この仕組みを導入することで最も重要な経験を学習させ、学習の収束につながると考えられる。しかし、メモリを増やすことで、計算効率が低下すると予測されるため、新たな工夫が必要であると考える。さらに結果の違いを明確にするため、追加実験として新しい環境も取り入れていきたいと考えている。

参考文献

[1] Volodymyr Mnih, Koray avukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller “Playing Atari with Deep Reinforcement Learning” arXiv:1312.5602 Thu, 19 Dec 2013

[2] Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, Will Dabney “Recurrent Experience Replay in Distributed Reinforcement Learning” Arcade Learning Environment, DQN Replay Dataset 21 Dec 2018, Last Modified: 06 May 2023

[3] Hado van Hasselt , Arthur Guez, and David Silver ”Deep Reinforcement Learning with Double Q-Learning” Proceedings of the AAAI Conference on Artificial Intelligence, 30(1). 2016-03-02

[4] Tom Schaul, John Quan, Ioannis Antonoglou, David Silver “Prioritized Experience Replay” arXiv:1511.05952 Thu, 25 Feb 2016

[5] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, Nando de Freitas ”Dueling Network Architectures for Deep Reinforcement Learning” arXiv:1511.06581 Tue, 5 Apr 2016