深層強化学習アルゴリズムの特徴解析

1　はじめに

近年, 深層学習をはじめとする機械学習の分野がめざましい発展を遂げている. 機械学習は大きく教師あり学習, 教師なし学習および強化学習の 3 つに分類される. その中で人工知能がゲームをプレイするタスクなどを中心に強化学習の様々なアルゴリズムが提案されている.

強化学習は与えられた問題の状態と行動, そして報酬を正しく定めなければ適切に問題を解くことは難しい. よって一見同じに見える問題であっても元の問題に外乱が加わるなど少し条件が変われば同じアルゴリズムでこれらの問題を解いたとしても異なる結果が得られる可能性がある. そこで今回の実験では強化学習の中でも特に有効と考えられている深層強化学習に着目し, その複数の手法について元の問題と外乱を加えた問題それぞれで解かせることによって, それぞれの手法の挙動について数値実験に基づき解析した.

2　要素技術

2.1　Q 学習

強化学習のアルゴリズムは大きく分けて 2 つ存在し 1 つは方策反復法と呼ばれ, もう 1 つは価値反復法と呼ばれる. Q 学習は価値反復法の典型的なアルゴリズムである. 価値反復法では成功時に報酬を与えて, 現在の状態とそれに対する行動の組に対して報酬に基づく価値を定める. 状態と行動の組によって定まる価値を行動価値関数または Q 値と呼ぶ. そして次ステップと現ステップの Q 値の差分である TD 誤差を計算してその差分に応じて現在の Q 値を更新する. この更新により状態に対する行動の価値を学習していき正しい行動を選択させる方法である. Q 値の更新式は以下の式で表される.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(R_{t+1} + \gamma \max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \tag{1}$$

ここで t は時刻, s は状態, a は行動, α は学習率, R は報酬, γ は時間割引率である. Q 学習において Q 値を何らかの形で格納する必要があり, 一般的には状態と行動をルックアップテーブルで表現する.

2.2　Deep Q-Network(DQN)

Q 学習で問題を解く場合, 問題の状態数が多かったりそもそも状態が連続値であると, 初めて直面する状態が多くなる. すると Q 値が適切に更新されず正しい行動を学習するために多くの時間がかかるという問題が発生する. さらにルックアップテーブルとして Q 値を格納するためにメモリを莫大に消費してしまうといった問題もある. これらの問題の対応策として連続値などを離散化してルックアップテーブルを実現する方法はあるが, 離散化する方法は人間によって定める必要があるため膨大な手間を要する.

これらの問題を解決するために DeepMind 社が 2013 年に発表した Deep Q-Network (DQN) [1] では Q 値をルックアップテーブルの代わりにニューラルネットワークによって表現する. つまり, 各状態変数の値を入力としてそれに対する行動ごとの Q 値を出力値として返すニューラルネットワークを作ることによって Q 学習の問題を解決した. Q 値の更新はニューラルネットワークの学習によって, 状態が連続値であっても適応する汎化能力を持つ. またゲーム画面のような画像を入力する状態とすることができるため, 学習する環境ごとに状態を手作業で設定する必要がない. DQN はスペースインベーダなどで知られる Atari のゲームの一部で人間のスコアを超える高い性能を示すなど有効性が評価されており, 提案以降 DQN の改良アルゴリズムが多く提案されている.

DQN には以下の 4 つの工夫がされている.

Experiment Replay

状態や行動などの組をメモリに格納してミニバッチ学習でメモリからランダムに取り出すことによって時間的に相関が高い内容を連続して学習することを防ぎ学習を安定させる.

Fixed Target Q-Network

DQN の Q 値の更新には (1) 式を見ればわかるように Q 値自身を使うため学習が安定しない傾向に

ある. そのため行動を決定する Q 値のネットワークと Q 値自身を更新するための Q 値のネットワークを分けることによって, 学習を安定させる.

Reward Clipping
すべての問題の報酬を -1, 0, 1 のいずれかにすることで同じハイパーパラメータで学習可能にする.

誤差関数
ネットワークの学習の際の誤差関数を 0 付近で微分可能かつ外れ値に敏感になりにくい Huber 関数を用いて学習を安定させる.

2.3 Rainbow

DQN によって Atari ゲームをプレイするような難度のタスクを解けるようになったが, それでも学習が安定しにくいといった問題が存在していた. これらの問題を解決するため DQN の改良手法が数多く提案されてきた. その代表的な改良手法である, Double-DQN(DDQN), Prioritized Experience Replay, Dueling Network, N-Step learning, Noisy Network, Distributional DQN の 6 手法と元々の DQN を併用する Rainbow と呼ばれる手法が 2018 年に発表された [2]. それぞれの改良手法に関する説明は紙面の都合上割愛する.

2.4 CartPole

CartPole は OpenAI Gym が提供している強化学習で利用できる環境の 1 つであり, 台車に乗った棒を倒れないように左右に動かしていく制御課題である. 状態はカートの位置, カートの速度, 棒の角度, 棒の角速度の 4 変数がそれぞれ連続値で表現されている. エージェントは各時刻ごとにこれらの 4 変数を観測することが可能である. CartPole では時刻のことをステップと呼び, より長いステップ間で制御し続けるほどエージェントの学習が進んでいるとみなす. またエージェントの行動は台車を左に動かすか右に動かすかの 2 種類である.

3 数値実験

3.1 問題設定

今回の実験では Rainbow に含まれるアルゴリズムのうち DQN, DDQN, Prioritized Experience Replay DDQN, Dueling Network DDQN, NoisyNet DDQN の 5 手法とこれらの手法をすべて搭載した手法の計 6 つに対して CartPole の環境を変えていき外乱を加えることで実験した. 元の CartPole では台車に加える力を 10 という値で固定して左右に動かしており, この力の値に平均が 0 の正規分布の値を加えることで行動における台車を動かす力に対してランダムな摂動を与えた. また標準偏差の値を変えることによって外乱の大きさを調整した. つまり今回の実験における台車を動かす力は以下の式で表される.

$$F = 10 + \epsilon \tag{2}$$

$$\epsilon \sim N(0, \sigma^2) \tag{3}$$

3.2 実験条件

表 1 に今回の実験で用いたパラメータを示す. 最適化手法については通常の CartPole 環境と標準偏差が 15 の正規分布に従う値を摂動として加えた環境のそれぞれで各アルゴリズムについて Optuna でパラメータを調整した.

また今回の実験ではエージェントが CartPole を 195 ステップ以上制御し続けたら成功として報酬を 1, 195 未満で棒が倒れたら報酬を -1, 倒れていなかったり, 200 ステップに達していない場合は報酬を 0 とした. 10 エポック連続で制御に成功したら学習自体が成功したとしてその時点で学習を打ち切り, 500 エポックに達しても 10 エポック連続で制御に成功していない場合は学習自体を失敗とした. 各アルゴリズムに対して 100 回実験をした. また簡単のためニューラルネットワークの入力は画像ではなく CartPole で設定されている前述の 4 状態を入力とした.

4 結果

4.1 実験 1

摂動の値が従う正規分布の標準偏差を 2, 3, 4, 5, 6, 10, 15 と増やしていき, 学習が完了した際の平均エポック数の推移を求めることでアルゴリズムの性能を調べた. 図 1 にこの結果を示す. 横軸は摂動の値が従う正規分布の標準偏差の大きさであり, 大きいほど環境に大きな外乱を加えていることを表す. 縦軸は学習が完了

表 1: 実験パラメータ

入力サイズ	4
隠れ層サイズ	32
出力サイズ	2
バッチサイズ	32
累積報酬の時間割引率	0.99
最大ステップ数	200
最大エポック数	500
最適化手法	AdaDelta もしくは Adam
損失関数	Huber

した際のエポック数であり, この値が小さいほど速く学習が完了しているのでアルゴリズムの性能が良いことを表す. 同図より DQN は外乱を加えていない環境でもあまり学習されず性能が高くないことが分かる. また外乱を強く加えていない環境では DDQN 単体が一番学習が早く進み逆に 5 つを全部のせしているアルゴリズムの方が遅い結果となった. 一方, 外乱を強く与えた環境では DDQN 単体, NoisyDDQN および 5 つを全部のせしたアルゴリズムが同程度の速さで学習していることが分かる.

4.2 実験 2

通常の CartPole 環境と力に対して標準偏差が 15 の正規分布で得られた値を加えて外乱を与えた環境の 2 つのそれぞれについて, エピソード数に対するステップ数の推移を求めた. 図 2 と図 3 にこの結果を示す. 横軸はエピソード数であり縦軸はステップ数である. グラフの上昇度合いが速いアルゴリズムほど立ち上りが速く学習されていることを表す. 図 2 より通常の CartPole 環境ではわずかながら NoisyNet や 5 つを全部のせしているアルゴリズムの方が立ち上りが遅くなっていることが確認された. しかし DQN 以外のアルゴリズムについては 100 エポックに到達する前には平均ステップ数が 175 以上となるほどには学習が速く進むことが分かった.

図 3 は図 2 と比べて DQN 以外の各アルゴリズムがエポックが経るにつれてステップ数が上昇しているものの, 局所的に上下振動している傾向が見られた. つまりどのアルゴリズムも外乱を強く与えることによって学習が不安定になっていることが分かる. また NoisyDDQN や 5 つを全部のせしたアルゴリズムが他のアルゴリズムと比べて立ち上りはかなり遅いものの, 300 エポックあたりから他のアルゴリズムの平均ステップ数を超え始め最終的には一番学習が進んでいる結果となった. この 2 つのアルゴリズムを比べると NoisyDDQN の方が学習が速く進むことが分かる. 一方 PrioritizedDDQN や DuelingDDQN は低エポックでは DDQN 単体と同速度で学習が進んでいる. しかし高エポックでは平均ステップ数が DDQN より下回り性能が悪い結果となった.

4.3 実験 3

標準偏差が 15 の正規分布で得られた値を加えて外乱を与えた環境について, 各アルゴリズムの学習失敗した数を求めた. 表 2 にその結果を示す. 同表より, DuelingDDQN や PrioritizedDDQN は他の手法と比べて外乱を強く与えた時に学習が失敗しやすいことが分かる. また NoisyDDQN や 5 つを全部のせしたアルゴリズムは他のアルゴリズムと比べて失敗した数が少ない結果となった. つまり NoisyNet を搭載している手法は外乱にも比較的強く学習が進んでいることが確認された.

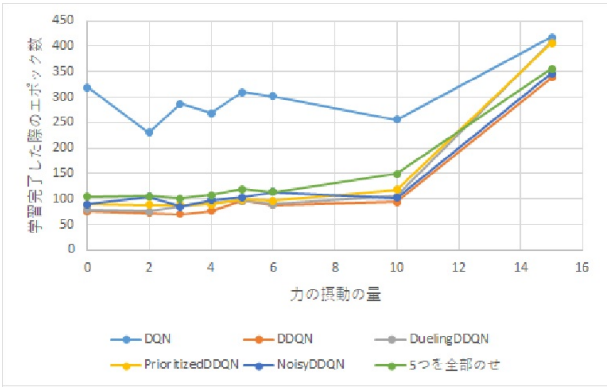

図 1: 摂動の度合いを変えていった際の平均終了エピソード数の推移

表 2: 外乱を加えた環境での学習失敗した数

アルゴリズム	学習失敗した数
DDQN	46
DuelingNetDDQN	57
PrioritizedDDQN	65
NoisyNetDQN	25
5 つを全部のせ	28

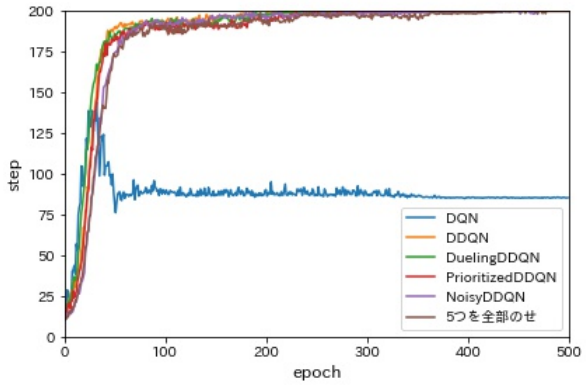

図 2: 通常の CartPole 環境におけるエピソード数に対する平均ステップ数の推移

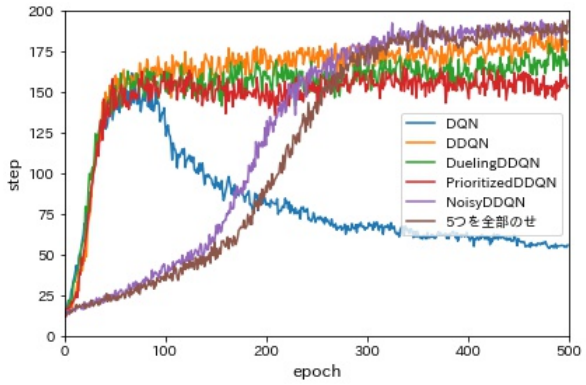

図 3: 標準偏差が 15 の正規分布に従う摂動を加えた CartPole 環境におけるエピソード数に対する平均ステップ数の推移

5 考察

実験 1 より外乱を強く与えていない環境では DDQN 単体のような多くのアルゴリズムを搭載していない手法の方が少し速く学習が進んだ. 一方, 外乱を強く与えた環境では NoisyNet を搭載した手法と DDQN 単体が同程度の速さで学習が進んでいる. 実験 2 より外乱を強く与えた環境下では NoisyNet を含んだ手法は他の手法と比べて学習スピードは遅いものの最終的には一番安定して学習が進むことが分かる. 一方 Prioritized DDQN や Dueling DDQN は外乱を与えた際に学習が安定しない結果が得られた. これは図 3 における DuelingDDQN と 5 つを全部のせした手法の低エポックにおける学習の速さの違いに影響を与えていると考えられる. また実験 3 からも NoisyNet により外乱が強い環境であっても安定して学習が進む結果が得られている.

これらの実験により外乱を強く与えた環境におけるエージェントの学習の安定性には NosiyNet が寄与していることが示唆される. NoisyNet は Q 学習で探索と活用のトレードオフのバランスを保つために用いられる ε-greedy 法に代えて, ネットワーク自体にノイズを加えてノイズの大きさ自体も学習していくアルゴリズムとなっている. NoisyNet は最終的な学習の向上に寄与するので, このアルゴリズムを含んだ手法が外乱に強く安定した学習をする結果となったと考えられる.

6 まとめと今後の課題

今回の実験では, CartPole 環境に外乱を与えた場合とそうでない場合のそれぞれについて, DQN や Rainbow に含まれる DQN の改良アルゴリズムに対する比較実験してそれぞれのアルゴリズムの特徴を解析した. その結果外乱を与えることによって学習が不安定になる傾向があり, また NoisyNet を含むアルゴリズムが少し学習速度は遅くなるが外乱に強く安定して学習することが分かった.

今後の課題としては Rainbow に含まれるアルゴリズムである N-step learning と Distributional DQN を比較実験できなかったので, これらのアルゴリズムを加えて実験することや CartPole 以外の学習環境に外乱を加えた環境でアルゴリズムの比較実験を行うことで, 各アルゴリズムの性能を評価することが考えられる.

参考文献

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013.

[2] Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Daniel Horgan, Bilal Piot, Mohammad Gheshlaghi Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *CoRR*, abs/1710.02298, 2017.