The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

経験データ重み付けによるDeep Q Networkの高速化

Accelerate Deep Q Network by weighting experiences

村上 知優 *1 Kazuhiro Murakami　森山 甲一 *1 Koichi Moriyama　武藤 敦子 *1 Atsuko Mutoh　松井 藤五郎 *2 Tohgoroh Matsui　犬塚 信博 *1 Nobuhiro Inuzuka

*1名古屋工業大学　大学院工学研究科　情報工学専攻
Department of Computer Science, Graduate School of Engineering, Nagoya Institute of Technology

*2中部大学　生命健康科学部　臨床工学科
Department of Clinical Engineering, College of Life and Health Sciences, Chubu University

Deep Q Network (DQN) is a reinforcement learning method using deep neural network to approximate Q-function. Literature tells us that DQN can choose better actions than humans. However, it needs much time to learn such actions. DQN learns its actions by using tuples of (state, action, reward, next state), called experiences, sampled from its memory. DQN samples them uniformly randomly, but the experiences are skewed. It results in slow learning because frequent experiences are redundantly sampled while infrequent ones hardly.

This work mitigates the problem by weighting experiences based on their frequency and manipulating their sampling probability. In a video game environment, the proposed method learned good actions faster than DQN.

1. はじめに

近年，工場における部品の取り付け作業や事務作業など，様々な場面で自動化が進められている．従来は人間が動作ロジックを全て定義することで行われてきたが，現実世界のより複雑なものを自動化するにあたって，動作ロジックを全て定義することは非常に難しく，現実的でない．こうした背景のもと，動作ロジック自体を機械が獲得する機械学習の分野が注目されており，中でも与えられた環境下で最適な行動を獲得する強化学習が大きな成果を挙げている [1]．また，近年画像認識や自然言語処理，音声認識などの分野で深層学習が大きな成果を挙げている [2]．深層学習は入力データから特徴を抽出することに長けており，人間には認識できないような特徴を抽出することができる可能性を秘めている．そして深層学習と強化学習を組み合わせた深層強化学習の手法として，Deep Q Network(DQN)[3][4] が提案された．DQN は Atari2600[5] のゲーム画面を入力とし，得られたスコアを報酬とすることで，上級者を上回るスコアを出せる行動を獲得するに至っている．しかし，深層強化学習は学習速度が遅く，その原因の一つに観測したデータ群（経験データ）の出現率が異なることが挙げられる．DQN は観測したデータ群（経験データ）を有限なメモリ内に保存し，そこから等確率でランダムにサンプリングした経験データを用いて学習を進めていくが，経験データはそれぞれ出現率が異なるため，出現率の低い経験データを重要な経験データとして扱い，サンプリングされる確率を操作することで学習効率を高めることができると考えられる．そこで本研究では，メモリ内の経験データそれぞれに重み付けを行うことで，DQN の学習速度の向上を図る．

連絡先: 村上知優，名古屋工業大学　大学院工学研究科　情報工学専攻，愛知県名古屋市昭和区御器所町，k.murakami.638@nitech.jp

2. Deep Q Network

2.1 Deep Q Network のアルゴリズム

Deep Q Network(DQN) は入力を状態 s，出力をその状態において選択することの出来る行動の行動価値とするディープニューラルネットワークである．本研究ではこのディープニューラルネットワークを「Q ネットワーク」と呼ぶことにする．経験データを蓄積するメモリを「ReplayMemory」といい，ReplayMemory からランダムに M 個サンプリングした経験データを用いて Q ネットワークの重みを更新する過程を「Experience Replay」という．NIPS 版 DQN[3] のアルゴリズムを図 1 に示す．図 1 にある f はエピソードの終端かどうかを表すフラグである．また，t は教師データを，r は報酬を表し，E は教師データとネットワークの出力の二乗誤差を表す．

Nature 版 DQN[4] ではこれに加えて学習を安定させるために次の 2 つの工夫を行う．本研究では Nature 版 DQN を扱い，本論文では単に DQN と表記されていた場合は Nature 版 DQN を指すものとする．

- ターゲットネットワークの導入
  教師データを生成する際に Q ネットワークの出力を用いている．しかし，Q ネットワークの重みは毎回更新されていくので，ある状態 s を入力した際の教師データが毎回変わってしまう．この問題を解決するために，ターゲットネットワークと呼ばれる Q ネットワークと全く同じ構造の別のネットワークを用意する．ターゲットネットワークの重みは毎回更新されず，Q ネットワークの重みが定期的にコピーされる．教師データ生成の際にターゲットネットワークの出力を用いることで，ある状態 s に対する教師データを固定し，学習を安定させる．したがって，教師データ生成の式を以下のように変更する．

$$t_i = \begin{cases} r_i & (f_i = True) \\ r_i + \gamma \max_{a'_i} Q_{target}(s'_i, a'_i) & (f_i = False). \end{cases}$$

  ここで，Q_{target} はターゲットネットワークの出力値を意味する．

- エラーのクリッピング
  図 1 の誤差の計算で，教師データと Q ネットワークの出力の差の部分（誤差信号）が二乗されている．これにより，誤差信号の絶対値が 1 より大きいか 1 以下であるかで誤差が大きく変化する．誤差が大きく変化するということは勾配が大きく変化するということであるから，学習が不安定になってしまう．これを防ぎ，学習を安定させる方法がエラーのクリッピングと呼ばれる手法である．以下の場合分けに従って新たな誤差信号 e_i' を計算し，誤差の計算は e_i' を用いて行う．

$$e_i = t_i - Q(s_i, a_i)$$

$$e_i' = \begin{cases} 1 & (e_i \geq 1) \\ e_i & (-1 < e_i < 1) \\ -1 & (e_i \leq -1). \end{cases}$$

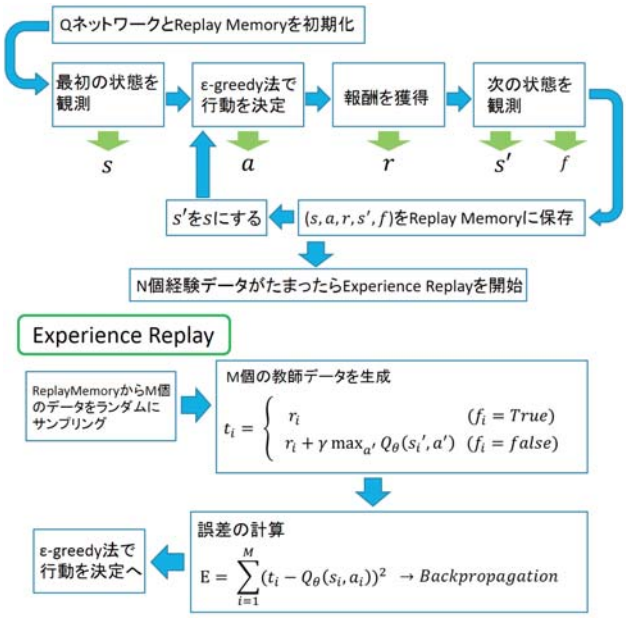

図 1: NIPS 版 DQN のアルゴリズム

2.2 Deep Q Network の問題点

Q ネットワークが最適な行動価値関数へ収束するためには出現する可能性のあるすべての経験データに対して十分な学習が必要であるが，出現率の低い経験データはサンプリングされる確率が低く，学習がなかなか進まないため，最適な行動価値関数への収束が遅くなってしまう．また，一度選ばれただけではその経験データに対する学習が十分に行われないため，出現率に応じて適度に選ばれることが好ましい．よって本研究では経験データそれぞれに重み付けを行い，出現率の低い経験データであるほどサンプリングされやすくすることでこの問題の解決を図る．

3. 提案手法

学習データに対して重み付けを行うことは機械学習の分野ではよく行われており，強化学習では Prioritized Sweeping[6] という手法が有名である．この手法は TD 誤差を指標として経験データに優先度を与え，キューに優先度順に保存し，優先度の高いものから順に経験データを取り出して学習に使用するというものである．本手法では経験データの重みを TD 誤差とは別の指標を用いて与え，かつ重みが大きいものが必ず学習に使用されるようにするのではなく，重みをもとに確率的に選択されたものが使用されるようにする．

DQN の構成に加えて新たに次の状態 $\boldsymbol{s}'$ を予測する A ネットワークを導入する．本手法では環境から与えられる状態がベクトルで表されるとする．図 2 に提案手法のネットワーク構成を示す．

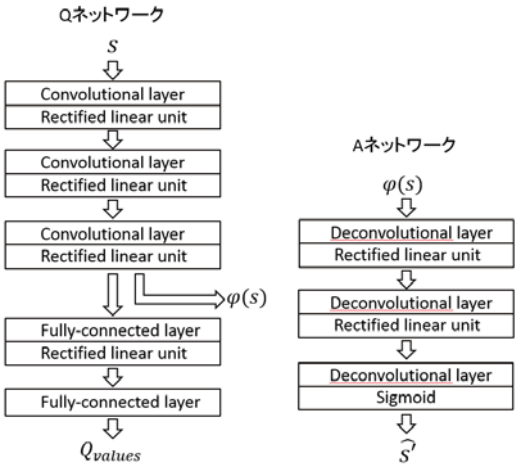

図 2: 提案手法のネットワーク構成

Q ネットワークの中間層の出力 $\boldsymbol{\varphi(s)}$ を A ネットワークへの入力とする．$\boldsymbol{\varphi(s)}$ は Q ネットワークの中間層の出力であるため，Q ネットワークの学習度合いと入力 $\boldsymbol{s}$ に応じて変化する．Q ネットワークに今までほとんど入力されていない状態 $\boldsymbol{s}$ が入力された場合，深層学習の特性 [8] から，$\boldsymbol{\varphi(s)}$ も今までほとんどとることのなかった値になり，A ネットワークの予測が大きく外れることになる．従って，A ネットワークの予測誤差を経験データの重みとすることで，出現率の低い経験データに大きな重みを与えることができる．

ReplayMemory からサンプリングした経験データの内 i 番目の重み $weight_i$ を次の式で計算する．

$$weight_i = \sum_j (s'_{i,j} - \hat{s'_{i,j}})^2.$$

ここで，$s'_{i,j}$ は i 番目の経験データに含まれる次の状態 $\boldsymbol{s}'$ の j 番目の要素である．そして $\hat{s'_{i,j}}$ は i 番目の経験データに対する A ネットワークの出力の j 番目の要素である．また，出現率の低い経験データによる学習が行われていくにつれて，その経験データに対する予測誤差は低下していくので，重みは徐々に小さくなっていくと考えられる．最終的に ReplayMemory 中の k 番目の経験データがサンプリングされる確率 P_k は，各経験データの重み W_k を用いて以下のように計算する．

$$P_k = \frac{W_k}{\sum_n W_n}.$$

A ネットワークの層数は予備実験の結果，3 層とした時が最も良い結果が得られた．また，Q ネットワークから $\boldsymbol{\varphi(s)}$ を取り出す場所は，勾配消失の関係上，出力層に近いほどその変動が

大きく望ましいが，全結合層の出力では入力の位置的な情報が失われてしまうため，A ネットワークでの状態 s' の予測が非常に困難になる．そこで本研究では，Q ネットワークの最も出力側の畳み込み層の出力をその場所とした．

4. 実験

4.1 環境設定

卓球を模したゲームである Atari2600 の Pong[5] を用いて実験を行った．エージェントが得点すると報酬 +1 が与えられ，逆に得点されると報酬 −1 が与えられる．これら以外は全て 0 の報酬が与えられる．エージェントの対戦相手は人間の記述した動作ロジック通りに動き，エージェントが適切な方向に球を打ち返さない限り対戦相手は失点しない作りになっている [5]．先に 21 点先取した方の勝利となり，この時 1 エピソード終了となる．エージェントのとることのできる行動は，上に移動，下に移動，その場で静止の 3 つである．また，どちらかに得点が入ると球は画面中央の初期位置に設置され，ランダムな方向に打ち出される．

この環境を選択した理由は，経験データの出現率に大きな違いがあるからである．特に学習序盤ではエージェントはほぼ無得点で負け続けることが予想されるので，+1 の報酬が得られた経験データの出現率が極めて低い．−1 の報酬が得られた経験データも 0 の報酬の経験データと比較すると出現率が低い．さらに，報酬 0 の経験データであっても，球が初期位置にある状態は得点あるいは失点後に必ず現れるため，出現率が他の状態に比べて高くなる．

本実験では，この環境に対して提案手法を適用することで，それぞれの経験データに対して出現率にもとづいた適切な重み付けができているか，学習速度が改善されるかを検証する．

予備実験の結果得られた最適なハイパーパラメータを表 1 に示す．ミニバッチサイズは ReplayMemory からサンプリングする経験データの個数 M で，ReplayMemory サイズは ReplayMemory に蓄えることのできる最大の経験データの個数である．Experience Replay 開始サイズはその個数の経験データを獲得した時点で Experience Replay を開始するということである．ターゲットネットワークの更新頻度はその回数だけ状態観測を行ったらターゲットネットワークを更新するということである．また，Q ネットワークの最後の全結合層の重みを全て 0 で初期化することで，Q ネットワークの学習が始まる前の出力値が全て 0 になるようにした．それ以外の層の重みは乱数による 0 から 1 の値で初期化した．使用した計算機は Intel Xeon E5-2650 v4 を 2 基，Nvidia GeForce GTX 1080Ti を 2 基持ち，メインメモリは 64GB である．

割引率 (γ)	0.99
ϵ の初期値	1.0
ϵ の最終値	0.1
ミニバッチサイズ (M)	32
ReplayMemory サイズ	100000
Experience Replay 開始サイズ (N)	10000
ターゲットネットワークの更新頻度	10000
Q ネットワークの学習率	0.00025
A ネットワークの学習率	0.00025

表 1: ハイパーパラメータ

4.2 実験結果

4.2.1 経験データの重みの可視化

図 3，図 4 は 100000 個の経験データ，図 5 は 16000000 個の経験データを獲得した時点での ReplayMemory 内の一部の重みの様子を可視化したものである．横軸が ReplayMemory 内の経験データのインデックス，縦軸が経験データの重みである．青色は報酬 0，橙色は報酬 −1，赤色は報酬 +1 の経験データを表す．また，経験データはインデックスが小さい方から時系列順に保存されており，図 5 の横軸は下 5 桁のみ表示されている．

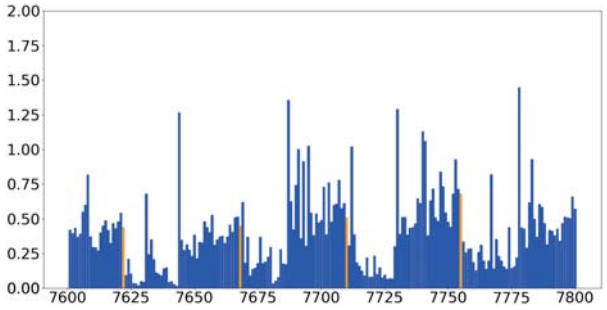

図 3: 開始から 7600〜7800 個の経験データ

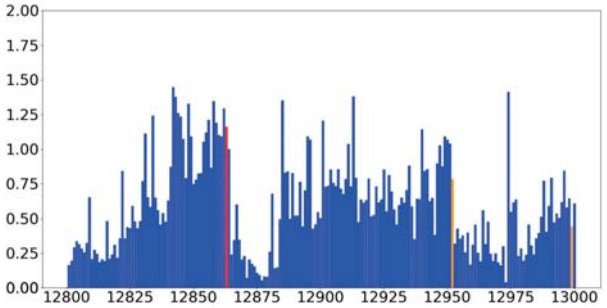

図 4: 開始から 12800〜13000 個の経験データ

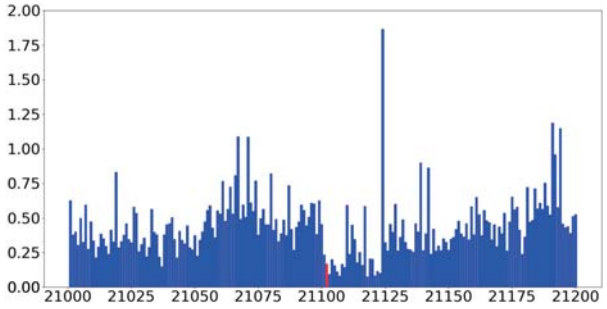

図 5: 開始から 15921000 個〜15921200 個の経験データ

図 3 には赤色の経験データがなく，橙色の経験データが 4 個存在する．これは，エージェントが 1 点も獲得することなく相手に 4 連続得点されているということである．また，一番左側以外の橙色の経験データがほぼ等間隔で並んでいることから，球は初期位置から右側のバー側へ打ち出され，そのまま打ち返すことができず，失点するということが 3 回連続で起こっ

ていると考えられる．さらに，−1 の報酬が得られた経験データに近づくにつれて経験データの重みが大きくなり，−1 の報酬が得られた経験データの後は重みが小さくなっている傾向があることがわかる．これは報酬 0 の経験データの中でも出現率が違うことが影響していると考えられる．また，図 4 より学習序盤において特に貴重な経験データとなる +1 の報酬が得られた経験データと，それが得られる過程の経験データに対して特に大きな重みが与えられていることがわかる．さらに図 5 より，出現率の低い経験データはいつまでも大きな重みではなく，重みが小さくなっていることがわかる．以上の結果より，それぞれの経験データに対して出現率にもとづいた適切な重み付けができていると考えられる．

4.2.2 合計報酬の推移

図 6 の横軸はこれまでにエージェントが学習中に観測したゲーム画面の合計フレーム数，縦軸はエージェントが獲得した報酬の合計値である．青色が DQN，緑色が提案手法のグラフである．

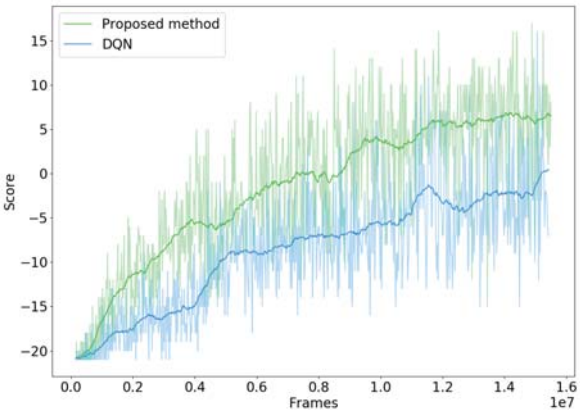

図 6: 合計報酬の推移

図 6 から，学習序盤で大きく差が開いていることが確認できる．DQN に比べて提案手法の方が少ないフレーム数で高いスコアを獲得できており，学習効率が改善されていることがわかる．また，移動平均が初めて 0 以上の値となるまでの実時間を比較すると，提案手法の方が約 32%高速化することができた．

4.3 Q 値の最大値の推移

図 7 の横軸はこれまでにエージェントが学習中に観測したゲーム画面の合計フレーム数，縦軸は Q ネットワークの出力したそれぞれの行動価値の最大値の 1 エピソード間の平均値である．行動価値の初期値が 0 で，学習率が小さいので，行動価値が徐々にしか上昇しない．よって，このデータが学習の進行を測る指標になる．青色が DQN，緑色が提案手法のグラフである．

図 7 から，DQN に比べて提案手法の方が早い段階で高い平均値となっている．よって，提案手法の方が報酬 +1 の経験データに対する行動価値関数の学習および，+1 の報酬を獲得する過程の状態行動対の組み合わせに対する行動価値関数の学習が優先的に行われていると考えられる．

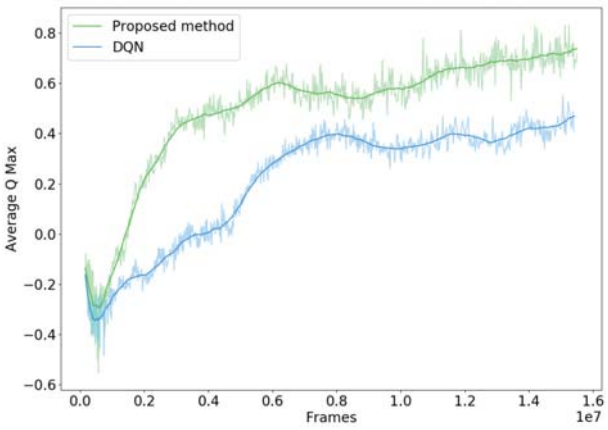

図 7: Q 値の最大値の推移

5. まとめ

本研究では DQN の経験データに対して出現率にもとづく重み付けを行うことで，学習速度を改善する手法を提案した．Atari2600 の Pong という経験データの出現率に大きな違いがある環境で実験した結果，経験データの出現率に基づいた動的な重み付けを行うことで，DQN に比べて少ない学習回数で高い性能を獲得することができ，学習速度を改善することができた．

今後の課題としては，経験データの出現率の違いの度合いが異なる環境での比較検証が挙げられる．

参考文献

[1] Pieter Abbeel et al.: Autonomous Helicopter Aerobatics through Apprenticeship Learning, *International Journal of Robotics Research*, pp. 1–31 (2010).

[2] Kaiming He et al.: Deep Residual Learning for Image Recognition, arXiv:1512.03385v1 [cs.CV] (2015).

[3] Volodymyr Mnih et al.: Playing Atari With Deep Reinforcement Learning, *NIPS Deep Learning Workshop* (2013).

[4] Volodymyr Mnih et al.: Human-level control through deep reinforcement learning, *Nature*, Vol. 518, pp. 529–533 (2015).

[5] Greg Brockman et al.: Openai gym. arXiv:1606.01540 (2016).

[6] Andrew W. Moore and Christopher G. Atkeson: Prioritized sweeping: Reinforcement learning with less data and less time, *Machine Learning*, Vol. 13, pp. 103–130 (1993).

[7] David Silver et al.: Mastering the game of Go with deep neural networks and tree search, *Nature*, Vol. 529, pp. 484–489 (2016).

[8] David E. Rumelhart et al.: Learning representations by back-propagating errors, *Nature*, Vol. 323, pp. 533–536 (1986).