動的ニューロンモデルを用いたニューラルネットワークへの因果トレースの適用

○山本一真[1] 柴田克成[1] （[1] 大分大学）

Application of Causality Traces to Neural Networks using Dynamic Neuron Model

∗Kazuma Yamamoto[1] and Katsunari Shibata[1] ([1]Oita University)

Abstract: Reinforcement learning using a neural network in a robot is expected to develop higher functions autonomously in the real world where a flood of sensor signals come without interruption. Causality traces that take in and hold the inputs according to the temporal change in the neuron's output have been proposed as a way that makes learning drastically faster and more efficient than in the case of eligibility traces. Effectiveness of causality traces for static model of neurons was confirmed in the previous study. Application of causality traces to a neural network with dynamic neuron model that is useful on forming complicated internal dynamics was attempted in this study. Analysis showed that the problem occurred was due to the lag of the output change behind the inputs in each neuron. To solve that, it is proposed that inputs are taken in the traces through the first-order lag with the same time constant as the neuron. The effectiveness compared with the case of eligibility traces is shown.

Key Words: causality traces, dynamic neuron model, neural network, reinforcement learning, eligibility traces

1 はじめに

近年、画像認識や音声認識の分野において、多層のニューラルネット (NN) を学習させる Deep Learning が高い認識能力を持つことで大きな注目を集めている [1)]。これは、NN が学習によって、人間には表現できないような抽象的な特徴を内部に獲得する素晴らしい能力を持っているためである。また最近では、Deep Learning と強化学習を組み合わせた Deep Q-network(DQN) を用いて 49 種類のゲームを学習させ、その半数以上で人間に匹敵するスコアを記録したという報告もある [2)]。NN を強化学習と組み合わせて学習させることによって、教師信号を与えなくても自律的に学習できる。さらに、その際、認識だけでなく、タスクの達成に必要な予測や、記憶などの機能が内部に創発することを筆者らのグループで示してきており、高次機能の創発という観点から今後の進展が大きく期待される [3)]。

実世界に住むわれわれは、センサから空間的にも時間的にも膨大な情報を得て、そこから必要な情報を抽出、生成することで適切な行動を行うことができる。Deep Learning が示す能力から、視覚からの空間的に膨大なセンサ信号の処理に NN が有効であることがわかる。しかし、これらのセンサ信号は刻々と変化していき、時間的にも膨大である。このような中でロボットの処理を学習させるためには、センサ信号を記憶し、それを用いて、過去に遡って学習することが考えられる。しかし、膨大なセンサ信号を全て記憶することは不可能である。また、適格度トレース (Eligibility Traces)[4)5)] という一種のメモリを用意し、新しいセンサ信号を取り込みつつ、記憶した情報を少しずつ忘却させていくことで、過去のセンサ信号を学習に利用する方法も用いられている。しかし、適格度トレースが取り込んだセンサ信号は、重要かどうかに関わらず、一定の減衰率で時間の経過とともに減衰してしまうため、減衰率が大きいと遠い過去の情報は消えてしまい、小さいとたくさんの過去の情報に埋もれてしまうという問題がある。

これに対して筆者らは、ニューロンの出力の変化が大きい時はよりその時の入力を記憶し、出力の変化が小さい時は記憶している情報を保持する因果トレース (Causality Traces) を提案した [6)7)]。これによって、出力の変化を引き起こす入力を選択的に記憶することができるため、過去の処理の学習を非常に効率的に行うことができる。

因果トレースはこれまで、一般的によく用いられる静的ニューロンモデルを用いた NN で使用され、その有効性が確認されてきた [6)7)]。一方、谷らの研究 [8)] などから動的ニューロンモデルは複雑なダイナミクスの形成に適していると考えられ、ロボットの知能の向上への寄与も期待できる。これらの理由から、本論文では因果トレースを動的ニューロンモデルを用いた NN に適用する。そして、まず、実際に適用した際に発生した問題についてその原因を追究し、改善方法を提案する。その後、他の学習方法との比較を行って因果トレースの有効性を検証する。

2 因果トレースの動的ニューロンモデルへの適用

2.1 因果トレース

因果トレースは、NN 内の各ニューロンの各結合部に 1 つ配置される。第 l 層目の j 番目のニューロンの i 番目の入力に対する因果トレース $c_{j,i,t}^{(l)}$ は微分方程式の形で、そのニューロンの出力の時間変化 $dx_{j,t}^{(l)}$ の絶対値とその時の入力 $x_{i,t}^{(l-1)}$ を用いて次のように計算される。

$$\frac{dc_{j,i,t}^{(l)}}{dt} = \frac{|dx_{j,t}^{(l)}|}{dt}\left(x_{i,t}^{(l-1)} - c_{j,i,t}^{(l)}\right) \tag{1}$$

この式から因果トレースの時間変化は出力の時間変化に比例して大きくなることがわかる。つまり、出力の変化が大きいと、その時の入力を大きくトレースに取り込み、小さいと過去の値を保持する仕組みになっている。

SY0007/15/0000-0136 ©2015 SICE

本研究では、状態価値を出力とする 3 層の階層型 NN で状態価値のみを学習させ、その学習に因果トレースを使用した。強化学習の状態価値の学習では、ステップ幅を Δt とすると、NN の出力層である第 L 層にある状態価値の出力 $x^{(L)}$ と割引率 γ から、現在の TD 誤差

$$\hat{r}_t \quad = \quad r_{t+\Delta t} + \gamma^{\Delta t}\, x^{(L)}_{t+\Delta t} - x^{(L)}_t \tag{2}$$

が計算される。ここで、報酬がない、つまり $r_{t+\Delta t} = 0$ のとき、時刻 t の理想出力は時刻 $t+\Delta t$ の出力に割引率 $\gamma^{\Delta t}$ かけたものとなる。したがって、現在の TD 誤差 $\hat{r}_t$ から過去の時刻 t' での状態価値を学習させるには、時間の経過分を割引いた、$\gamma^{t-t'}\,\hat{r}_t$ で学習させる必要がある。さらに、時刻 t' において、当該ニューロンが状態価値の出力ニューロンへの貢献度を表わす感度

$$s^{(l)}_{j,t'} \quad = \quad \frac{\partial x^{(L)}_{t'}}{\partial u^{(l)}_{j,t'}} \tag{3}$$

の情報が時刻 t での学習時に必要となる。この感度は、誤差逆伝搬での誤差信号同様にニューラルネット内を逆伝搬させて計算することができる [6)7)]。

これらを考慮した上で、強化学習での状態価値の学習に使用する際の因果トレース $c^{(l)}_{j,i,t}$ の時間変化を離散時間で表現すると、次のようになる。

$$c^{(l)}_{j,i,t} \quad = \quad \gamma\,(1 - |\Delta x^{(l)}_{j,t}|)\, c^{(l)}_{j,i,t-\Delta t} + |\Delta x^{(l)}_{j,t}|\, s^{(l)}_{j,t}\, x^{(l-1)}_{i,t} \tag{4}$$

$$|\Delta x^{(l)}_{j,t}| \quad = \quad \frac{1}{2}\,(|x^{(l)}_{j,t+\Delta t} - x^{(l)}_{j,t}| + |x^{(l)}_{j,t} - x^{(l)}_{j,t-\Delta t}|) \tag{5}$$

式 (4) の第 1 項がトレースに保持していた情報の減衰であり、第 2 項が出力の変化に伴う情報の取り込みを表している。そして、重み値の更新を次のように行う。

$$\Delta w^{(l)}_{j,i,t} \quad = \quad \eta\,\hat{r}_t\, c^{(l)}_{j,i,t} \tag{6}$$

適格度トレースでは、式 (4) の因果トレースの減衰と取り込みの割合である $(1-|\Delta x^{(l)}_{j,t}|)$, $|\Delta x^{(l)}_{j,t}|$ の部分が λ, $(1-\lambda)$ と定数であるため、どのニューロンも一様にかつ一定の割合で過去の情報を取り込んで減衰していくことになる。しかし、因果トレースでは、出力の変化を引き起こす重要な入力を選択的に記憶、保持するとともに、ニューロン毎に出力の変化量が異なるため、各ニューロンで異なったタイミングの入力の保持が可能になり、それによってニューロン間の役割分担が加速することも示されている [6)7)]。さらに、学習が進むことで、時間軸上の重要なイベントに対してニューロン間で分担的に反応することも期待される。

2.2 動的ニューロンモデルへの適用

静的ニューロンモデルでは現在の入力によって現在の出力が一意に決まる。一方で、動的ニューロンモデルではニューロンの内部状態 $u^{(l)}_{j,t}$ は

$$\tau^{(l)}_j \frac{du^{(l)}_{j,t}}{dt} \quad = \quad -u^{(l)}_{j,t} + \sum_{i=1}^{N^{(l-1)}} w^{(l)}_{j,i}\, x^{(l-1)}_{i,t} \tag{7}$$

という微分方程式に基づいて計算され、時定数 $\tau^{(l)}_j$ に応じて過去の入力が現在の出力に影響してくる。これにより、ニューロン自体がダイナミクスを持つ。これを簡単のため差分方程式で近似し、

$$u^{(l)}_{j,t} \quad = \quad (1 - \frac{\Delta t}{\tau^{(l)}_j})\, u^{(l)}_{j,t-\Delta t} + \frac{\Delta t}{\tau^{(l)}_j} \sum_{i=1}^{N^{(l-1)}} w^{(l)}_{j,i}\, x^{(l-1)}_{i,t} \tag{8}$$

と計算する。第 1 項が前の時刻の内部状態 $u^{(l)}_{j,t-1}$ を $(1-\frac{\Delta t}{\tau^{(l)}_j})$ で減衰させ、第 2 項で減衰させた分 $\frac{\Delta t}{\tau^{(l)}_j}$ だけ現在の入力とニューロンの結合の重み値の積和の値を取り込んでいる。$\frac{\Delta t}{\tau} = 1$ とすると、静的ニューロンモデルの式と一致する。

動的ニューロンモデルを用いた NN で状態価値の学習をさせるには、現在の出力に過去の入力が影響するため、過去の入力まで時間を遡って誤差信号を逆伝搬しなければならなかった。しかし、因果トレースはそのニューロンが現在の出力になるまでの時間変化を引き起こした入力を選択的に保持している。したがって、これを使うことで、時間を遡ることなくニューロンの出力に強く影響した過去の入力に対する学習ができると期待される。そこで、動的ニューロンモデルの場合でも、静的ニューロンモデルと同じように式 (3) から式 (6) を用いて因果トレースの更新と重み値の更新を行った。

3 タスク設定

因果トレースが動的ニューロンモデルでも動作するかどうかを確認するために、先行研究 [7)] でも使用した、因果トレースの効果を観察しやすいように設計されたタスクを用いた。Fig. 1 に示すように、100 個の領域に分割された 1 次元のフィールドがあり、奇数番目の領域は通過に長時間 (200step) かかる領域、偶数番目の領域は短時間 (2step) で通過できる領域とした。長時間滞在する領域と短時間滞在する領域が混在しているため、一定時間毎に入力を取り込む適格度トレースでは短時間領域の学習が困難になる。一方、因果トレースならば、短時間領域でも領域の境界部で入力が大きく変化し、結果的に中間層ニューロンの出力が大きく変化するため、その時の入力を大きく取り込むことができ、学習が容易になる。

ここでは簡単のため、エージェントはフィールドの左端をスタートし、探索 (試行錯誤) を行わずに領域を常に一定の速さで右向きに移動し、ゴールに着くと報酬を得る。そして、3 層 NN の出力を状態価値として、入力からその

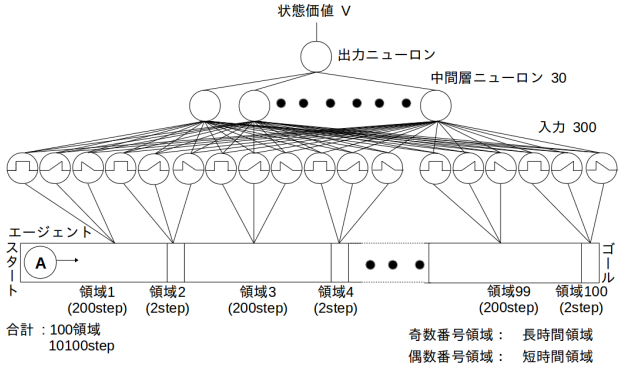


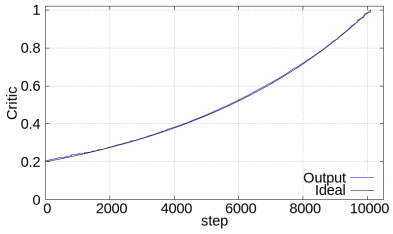
Fig. 1 不均一な領域からなる一次元の移動タスクにおける状態価値 (critic) の学習

状態のゴールまでの近さを学習させる。入力信号は各領域に 3 種類ずつの合計 300 個ある。それぞれの信号は、当該領域にいる間 1 の一定値をとるもの、領域中の位置を与えるために領域中で 0〜1 まで一定の割合で連続的に増加するものと、領域中で 1〜0 まで一定の割合で連続的に減少するものの 3 種類があり、エージェントが当該領域にいない場合はどの入力も 0 の値をとる。このように、領域毎に局所的に反応する入力信号を用意することで、領域が変わると入力が大きく変わるため、NN の汎化能力が効かず、トレースによる学習の効果が見やすくなる。

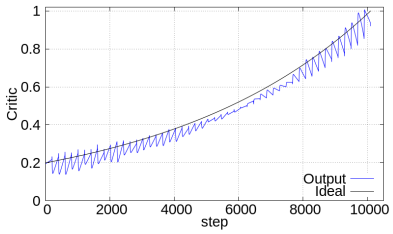
先行研究では、このタスクを用い、静的ニューロンモデルで因果トレースと適格度トレースで時定数をいくつか変化させた場合との比較を行い、因果トレースを用いた場合は、30 試行ほどで、200step の領域、2step の領域の区別なく学習でき、その優位性を確認した [6)7)]。

4　学習結果と問題点の解析

因果トレースを静的ニューロンモデル ($\frac{\Delta t}{\tau} = 1$) で用いた場合と $\tau = 4$ とした動的ニューロンモデル ($\frac{\Delta t}{\tau} = \frac{1}{4}$) で用いた場合の 30 試行学習した後の各ステップでの状態価値の出力を、割引率から計算した理想価値と比較したものを Fig. 2 に示す。また、それぞれの学習曲線を Fig. 3 に示す。Fig. 3(a)(b) の横軸は試行回数で、(a) の縦軸は理想価値と NN の出力との差の絶対値の平均であり、主に全体的な誤差を表し、理想曲線に出力が近づく早さを表している。(b) の縦軸は、各ステップでの TD 誤差の絶対値の平均であり、局所的な誤差を表し、状態価値に Fig. 2(b) のようなギザギザがあると局所的な誤差が大きくなり、主に状態価値の曲線の滑らかさを見ることができる。



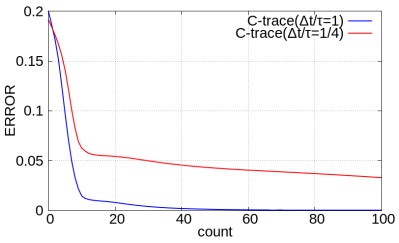
(a) C-trace($\frac{\Delta t}{\tau} = 1$)



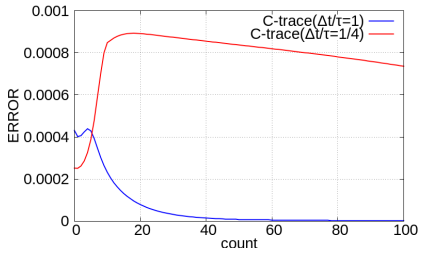
(b) C-trace($\frac{\Delta t}{\tau} = \frac{1}{4}$)

Fig. 2　30 試行目の状態価値の出力の比較

Fig. 2(a) の因果トレースを静的ニューロンモデルで用いた場合は理想曲線とほぼ重なっており、Fig. 3(a) の理想曲線との差も Fig. 3(b) の TD 誤差も、学習の最初の方ですぐに減少していることが確認できる。しかし、Fig. 2(b) の動的ニューロンモデルで用いた場合は出力にギザギザした形が見られた。このため、Fig. 3(a) を見ると 10 試行を越えたあたりから偏差を減らす速度が遅くなり、その後

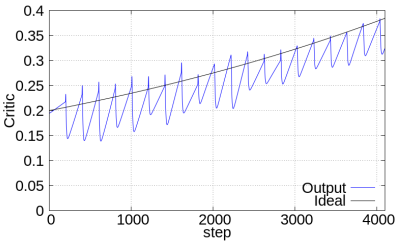


(a) 理想価値からの平均偏差

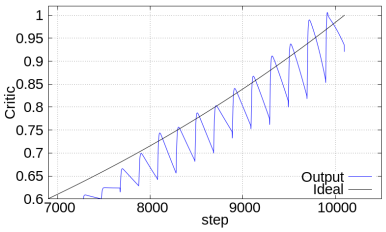


(b) TD 誤差の絶対値の平均

Fig. 3　静的ニューロンモデルの場合との学習曲線の比較



(a) スタート付近の拡大図

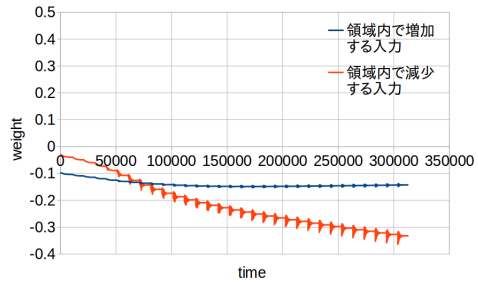


(b) ゴール付近の拡大図

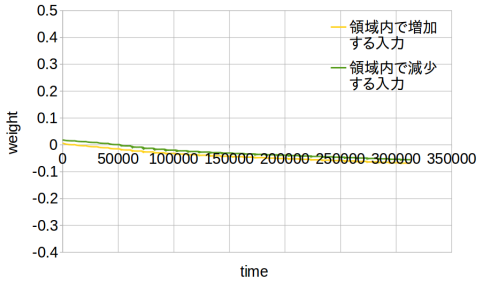
Fig. 4　C-trace($\frac{\Delta t}{\tau} = \frac{1}{4}$) の 30 試行目の状態価値の出力の拡大図

もギザギザが消えず、偏差が残り続けている。また、Fig. 3(b) では 10 試行目あたりまでギザギザが発生することで逆に誤差は大きく上昇しており、その後、少しずつ減っていくことが確認できるが、静的ニューロンモデルで用いた場合と比べると学習がかなり遅いことがわかる。

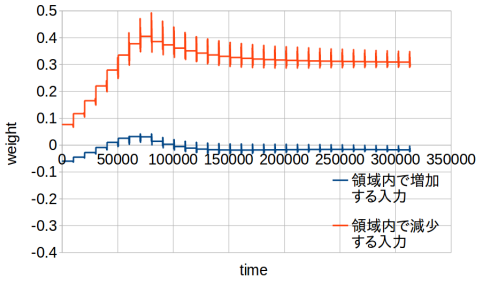
Fig. 4 は Fig. 2(b) のスタート付近とゴール付近をそれぞれ拡大したものである。Fig. 4(a) では右上がりのギザギザ、(b) では右下がりのギザギザが見られる。Fig. 5 に、長時間領域と短時間領域のそれぞれで、ある中間層ニューロンのスタート付近の入力と結びつく重み値とゴール付近の入力と結びつく重み値の学習による変化の様子を示す。Fig. 5 ではスタートからの step 数×試行回数を横軸に取り、重み値の変化を示す。1 試行が 10,100step であるため、10100step の周期的な変化が見られる。また、Fig. 5(a)(b)



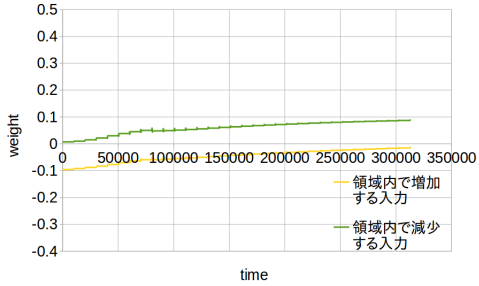
(a) スタート付近の長時間領域の入力と結びつく重み値



(b) スタート付近の短時間領域の入力と結びつく重み値



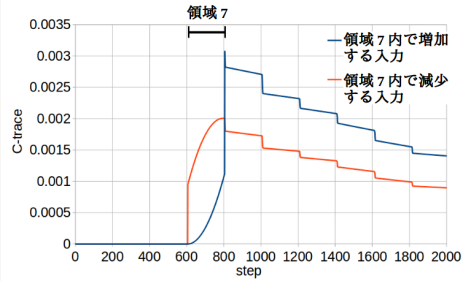
(c) ゴール付近の長時間領域の入力と結びつく重み値



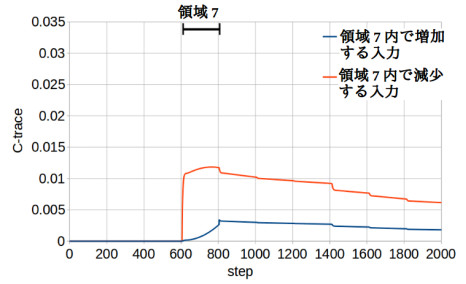
(d) ゴール付近の短時間領域の入力と結びつく重み値

Fig. 5　中間層 29 個目のニューロンの重み値の初期重み値からの時間変化

では、スタート付近で入力が入り、それが因果トレースで保持されるため、その後も重み値の変化が継続する。一方、Fig. 5(c)(d) では、入力が入るのはゴール付近であるため、重み値の変化はゴール付近でのみ起こる。Fig. 5(a) を見ると、“領域内で減少する入力” と結びつく重み値が “領域内で増加する入力” と結びつく重み値と比べ、大きく更新されており、同じ長時間領域である Fig. 5(c) を見ても同じ傾向にある。短時間領域の Fig. 5(b)(d) では、どちらの入力も同じように変化しており、その変化はあまり大きくない。



(a) 時定数 $\frac{\Delta t}{\tau} = 1$



(b) 時定数 $\frac{\Delta t}{\tau} = \frac{1}{4}$

Fig. 6　30 試行目の中間層 29 個目のニューロンの長時間領域の入力を取り込む因果トレースの時間変化

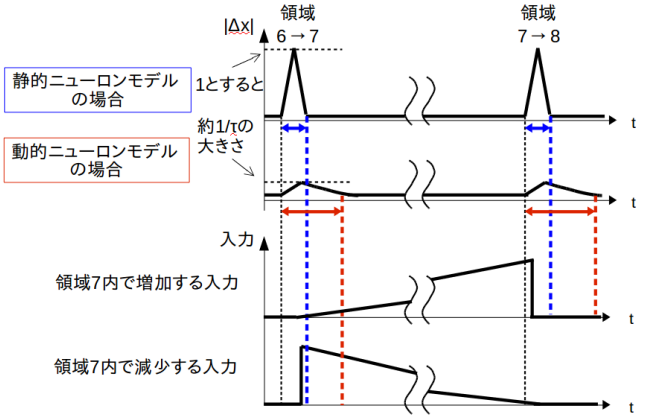


Fig. 7　ニューロンの時定数による因果トレースへの入力の取り込みの違い

そこで、長時間領域での 2 種類の入力に対する重み値の変化の差の原因を探るため、重み値の更新式に含まれる因果トレースが長時間領域の入力をどのように保持、減衰しているかを確認した。Fig. 6 は、スタートに近い長時間領域である領域 7(606〜805step) の前後での因果トレースの値の変化を静的と動的のニューロンモデルで比較したグラフである。領域 7 に入る時と出る時に入力は大きく変化し、ニューロンの出力も大きく変化する。したがって、因果トレースはこの時の入力を大きく取り込むことになる。Fig. 6 の (a) と (b) を比較すると、領域 7 から出る時の “領域内で増加する入力” の因果トレースへの取り込みが大きく異なることがわかる。この違いを Fig 7 を使って説明していく。

まず、静的ニューロンモデルでの因果トレースでは、

Fig. 7 から出力の変化が領域 6 と 7 の変わり目、7 と 8 の変わり目でのみ大きくなっていることで領域 7 の “領域内で増加する入力” と “領域内で減少する入力” は、同じようにトレースに取り込まれる。このため、Fig. 6(a) の静的ニューロンモデルでの因果トレースはどちらの入力も領域の変わり目でトレースが大きく取り込んでいる。しかし、動的ニューロンモデルでの因果トレースでは、Fig. 7 のように出力の変化が緩やかであるため、出力の変化量が $\frac{\Delta t}{\tau} = 1$ に比べ、大きさが約 $\frac{1}{4}$ と小さくなり、横に引き伸ばされた形となって、その分長い時間かけて少しずつ入力を取り込むことになる。

領域 7 に入る時は、その後しばらくの間 “領域内で減少する入力” は大きな値を取るため、トレースに十分に取り込まれる。一方、領域 7 から出る時は、“領域内で増加する入力” がすぐに 0 に落ちてしまうため、十分にトレースに取り込むことができない。これにより、Fig. 6(b) の動的ニューロンモデルでの因果トレースは “領域内で減少する入力” を取り込むトレースは大きくなり、“領域内で増加する入力” を取り込むトレースの変化はかなり小さくなっている。

Fig. 5(a)(c) で長時間領域の “領域内で減少する入力” と結びつく重み値が、“領域内で増加する入力” と結びつく重み値と比べて大きく更新されている原因は、トレースが “領域内で減少する入力” は大きく取り込み、“領域内で増加する入力” をほとんど取り込めていないことだとわかる。さらに、スタート付近では入力が入ると状態価値が下がるように学習するため、“領域内で減少する入力” によって、Fig. 4(a) のように、出力は長時間領域内で右上がりに増加する。一方、ゴール付近では入力が入ると状態価値を上げるように学習するため、“領域内で減少する入力” によって Fig. 4(b) のように、右下がりのギザギザになることがわかる。つまり、時定数が大きなニューロンの出力の変化は緩やかであるのに対し、トレースに取り込む入力の変化は急であることで入力の取り込みのタイミングがずれていることが原因と考えられる。

5 動的ニューロンモデルに適した因果トレースの提案

前述の因果トレースが入力を取り込む時間のずれの問題を解決するため、トレースに取り込む入力 $x_{i,t}^{(l-1)}$ を

$$\tilde{x}_{i,t}^{(l-1)} = (1 - \frac{\Delta t}{\tau_j^{(l)}})\, \tilde{x}_{i,t-1}^{(l-1)} + \frac{\Delta t}{\tau_j^{(l)}}\, x_{i,t}^{(l-1)} \qquad (9)$$

とニューロンと同じ時定数の一次遅れ系を通すことによって、Fig. 8(b) のように保持、減衰させる方法をとった。そして、それを因果トレースに

$$c_{j,i,t}^{(l)} = \gamma\,(1 - |\Delta x_{j,t}^{(l)}|)\, c_{j,i,t-1}^{(l)} + |\Delta x_{j,t}^{(l)}|\, s_{j,t}^{(l)}\, \tilde{x}_{i,t}^{(l-1)} \qquad (10)$$

と取り込むようにした。これによって、長時間領域の “領域内で上昇する入力” の取り込みが増え、“領域内で下降する入力” の取り込みが抑えられるため、両入力の取り込みが大きく異なる問題が改善されると期待される。

6 学習結果と適格度トレースとの比較

因果トレースを静的ニューロンモデル ($\frac{\Delta t}{\tau} = 1$) に適用した場合、動的ニューロンモデル ($\frac{\Delta t}{\tau} = \frac{1}{4}$) に適用した場合、さらには、比較のために適格度トレースを動的ニューロンモデルに適用し、その減衰を表わす λ の値を 0.999 とした場合と 0.99 とした場合の 4 通りについて、30 試行学習した後の状態価値を Fig. 9(a) から (d) に示す。また、それぞれの学習曲線を Fig. 10 に示す。ただし、動的ニューロンモデルに因果トレースを適用するにあたり、もう 1 つの問題点として、短時間領域では入力が 2step しか入らないのにニューロンの時定数が大きくてニューロンが入力の情報を十分に取り込めないというタスク設定上の問題点があることがわかったため、ここでは $\tau = 4$ とせず、$\tau = 1$ のままで $\Delta t = \frac{1}{4}$ とし、同一環境を 40,400step かけてゴールに到達するようにした。また、これに伴い、割引率は $\gamma^{\frac{1}{4}}$ となり、1step あたりの状態価値の変化量が小さくなるため、Fig. 10(b) の TD 誤差の平均値は Fig. 3(b) より小さくなっている。

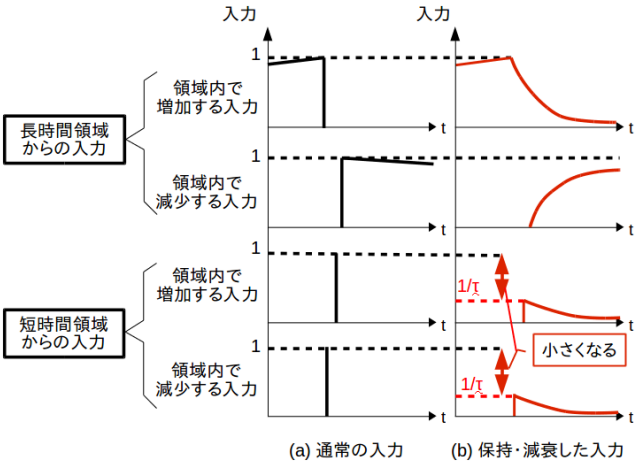
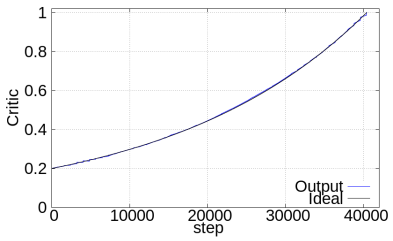


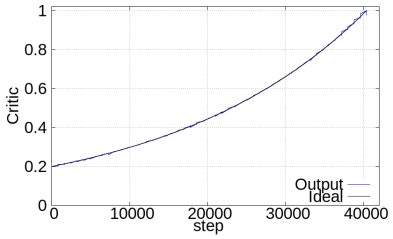
Fig. 8 トレースに取り込む入力を保持・減衰させた信号

Fig. 9(b) の因果トレースを動的ニューロンモデルに適用した場合では、トレースに取り込む入力を保持、減衰させることで右上がり、右下がりのギザギザは見られなくなり、Fig. 9(a) の因果トレースを静的ニューロンモデルに適用した場合と同様に、理想曲線と重なるように状態価値を学習することができている。Fig. 10 の (a) と (b) の 5 試行目辺りで、偏差も TD 誤差も若干大きくなっているが、これは、学習初期にわずかなギザギザが発生しているためであった。しかし、Fig. 10 の (a) では、20 試行を越えた辺りで偏差をほぼなくすことができており、Fig. 10(b) では、若干 TD 誤差が残ってはいるものの、静的ニューロンモデルに適用した場合の学習能力とほぼ変わらないことが確認できる。

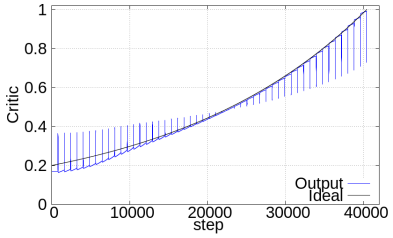
次に適格度トレースの場合を見ていく。Fig. 9(c) の $\lambda = 0.999$ の場合は評価の全体的な形は理想曲線とほぼ重なっているため、Fig. 10(a) では早い段階で偏差が小さくなっている。しかし、λ が大きいため、短時間領域の入力があまりトレースに取り込まれないことで学習が遅くなり、周期的なパルス列が見られる。このため、Fig. 10(b) では、TD 誤差がスタート付近で大きくなってから、あまり減っていないことがわかる。Fig. 9(d) の $\lambda = 0.99$ の場合は $\lambda = 0.999$ の場合より λ が小さくなったことで、パルス列が小さくなったが、30 試行目ではまだゴールから遠いところの学習ができていない。このため、Fig. 10(b) のよう



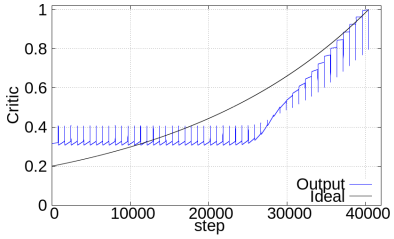
(a) C-trace($\frac{\Delta t}{\tau} = 1$)



(b) C-trace($\frac{\Delta t}{\tau} = \frac{1}{4}$)



(c) E-trace($\lambda = 0.999$, $\frac{\Delta t}{\tau} = \frac{1}{4}$)



(d) E-trace($\lambda = 0.99$, $\frac{\Delta t}{\tau} = \frac{1}{4}$)

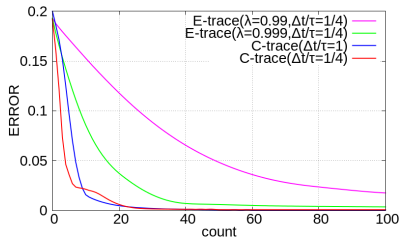
Fig. 9　30 試行目の状態価値の出力の比較

に、TD 誤差は $\lambda = 0.999$ の場合より小さいが、Fig. 10(a) のように理想価値との偏差の減り方が遅いことがわかる。

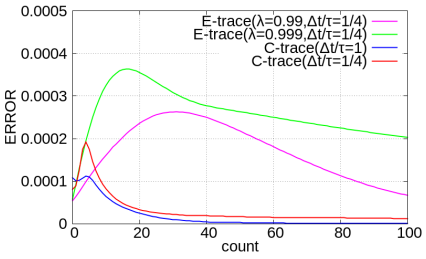
以上より、動的ニューロンモデルに因果トレースを適用した場合、ゴールから離れた状態まで素早く学習が進むとともに、短時間領域の学習も早く、適格度トレースを使用した場合と比較して学習性能が高いことが確認できた。この結果は、静的ニューロンモデルと近いものであった。

7　まとめ

本論文では、動的ニューロンモデルを用いた NN へ因果トレースを適用した場合の有用性を検証することを目的とした。静的ニューロンモデルと全く同じ方法で動的ニューロンモデルに適用したところ、動的ニューロンモデルではニューロンの出力が緩やかな変化であるのに対し、トレースに取り込む入力信号が急に変化していることが原因で、状態価値がうまく学習できなかった。しかし、取り込む入力をニューロンの時定数と揃えて保持・減衰させ、出力の



(a) 理想価値からの平均偏差



(b) TD 誤差の絶対値の平均

Fig. 10　因果トレースへの入力の取り込み方を変更した後の学習曲線の比較

変化に合わせるように取り込む入力の変化を緩やかにすることで、問題は解決された。静的ニューロンモデルを用いた NN で因果トレースを用いた場合と比べて、学習性能に大きな差がないことを示した。さらに、静的ニューロンモデルの場合と同様に、適格度トレースの場合に対する有用性を示すことができた。

動的ニューロンモデルは、RNN で使用することで複雑なダイナミクスを形成することが期待されるため、今後は RNN で因果トレースを使用して強化学習を行うことが大きな課題である。

参考文献

1) A. Krizhevsky, et al. : ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25, pp.1097-1105, 2012
2) V. Mnih, et al. : Playing Atari with Deep Reinforcement Learning, NIPS Deep Learning Workshop, 2013
3) 柴田 克成 : 強化学習とニューラルネットによる知能創発, 計測と制御, 第 48 巻, 第 1 号, pp.106-111, 2009
4) R. S. Sutton, A. G. Barto : Reinforcement Learning, pp. 163-192. MIT press, 1988
5) B. Bakker et al. : A Robot that Reinforcement-Learns to Identify and Memorize Important Previous Observations, Proc. of IROS 2003, pp.430-435, 2003
6) K. Shibata : Causality Traces for Retrospective Learning in Neural Networks - Introduction of Parallel and Subjective Time Scales, Proc. of IJCNN 2014, pp.2268-2275, 2014
7) 柴田 克成 : 因果トレース - 並列かつ主観的時間スケールの導入による過去の事象の効率的学習 - , 信学技法, NC2013-115, pp.157-162, 2014
8) 谷 淳 : ロボットで「科学」する記号の問題, 日本ロボット学会誌, 28 巻, 第 4 号, pp.522-531, 2010 年