

勾配法による動的なニューロンモデルの時定数の学習

大分大学 松木俊貴 柴田克成

Learning time constant of dynamic neuron model with gradient descent

Toshitaka Matsuki and Katsunari Shibata, Oita University

Abstract: In this paper, we propose a learning method to update the time constant in each dynamic neuron model to generate desired output patterns with gradient descent. Selecting appropriate time constants for each neuron in continuous-time recurrent neural network is difficult. Updating time constants based on gradient descent is also difficult because the sensitivity of the update is significantly varied depending on the present time constant value. Therefore, to avoid the instability, we propose a learning method applying gradient descent to the logarithm of the time constant. We carried out an “oscillator reproducing task” in which a network is trained to generate oscillatory outputs based on the outputs from a teacher network. The training result shows that our proposed method can successfully update the time constants and suggests that leaning of time constants expands the freedom of learning and improve the learning performance.

1 序論

近年、Deep Learning(DL)の学習能力を用いた成果が、様々な分野の研究で報告され大きな注目を集めている [1]。DLでは、入力から出力までの全体の処理が、重み値の修正によって大規模なニューラルネットワーク (NN) の中に獲得される。実世界には、空間の広がりや時間の流れが存在する。入力シーケンスの処理、情報の記憶や想起、連続的な運動出力の生成、意思決定や思考などを目指した今後の NN の発展のためには空間的な広がりだけでなく時間的な流れをいかに扱うかが重要な鍵となる。内部にフィードバックを持ち、情報が時間的に保持されるリカレント NN(RNN) は時間的な情報処理の学習に用いられる。DL で一般的な離散時間ニューロンモデルのように時間的な連続性を無視せず、内部状態のダイナミクスを微分方程式で表した動的なニューロンモデルがある。この動的なニューロンで構成される RNN は continuous-time recurrent neural network (CTRNN) と呼ばれる。このニューロンは、内部状態を減衰させつつ、そこに入力を取り込んで内部状態の更新を行う。また、時間スケールは時定数によって決まる。小さい時定数を持つニューロンは、更新の際に内部状態を大きく減衰させ大きく入力を取り込む。一方、大きい時定数を持つニューロンは、更新の際に内部状態をわずかに減衰させて小さく入力を取り込む。

ロボットが実環境の中で適切に行動を行うためには、様々な時間スケールにおける物事の時間的な関係を考慮しなければならない。例えば、ロボットがボトルに入ったジュースをグラスに注ぐケースを考える。短い時間スケールでは、ロボットは状況に応じて上手く迅速に運動指令を変化させていく必要がある。一方もう少し長い時間スケールにおいては、ボトルを持つ、キャップを外す、ジュースを注ぐ、グラスを手渡す、などといったような複数の行動を段階的に切り替えなければならない。このような複数の時間スケールの振る舞いを実現することは、離散時間ニューロンによって構成される RNN にとって非常に難しい。なぜなら、そのような RNN の時間スケールは計算処理の時間間隔によって決まってしまうため、一つの時間スケールしか持つことができないからである。複

数の時間スケールを一つのネットワークに導入するためには、異なる時定数を持つ動的なニューロンを用いることが有用である。

山下らは三つの固定された時定数による複数の時間スケールを持つ CTRNN に学習によって機能の階層性が獲得されることを示した [2]。しかし、巨大な並列処理システムである NN の中の全てのニューロンの時定数を予め最適な値に設定することは困難である。M.C.mozer は複数の時間スケールを持つ RNN が効果的に時間方向に展開される関数を近似できることを示し、同時に時定数の学習の持つ可能性とその難しさについて言及している [3]。

我々は、時定数を学習できれば、複数の時間スケールに対する機能の階層性が外部からの指示なく必要に応じてニューロン間に創発することや、学習の時間軸方向への拡張が学習パフォーマンスを向上させることなどができると期待している。そこで、ニューロンの時定数を重み値と同様に勾配法によって更新できないか考えた。しかし、時定数が小さい時には時定数の変化が非常に大きくしばしば時定数が負の値になってしまうことや、時定数の変化に対してニューロンの出力が非常に敏感であること、また一方で、時定数が大きい時には時定数の学習による変化が小さくなり、時定数の変化に対して出力も鈍感になることなどの問題により、勾配法によって時定数を直接的に学習することは非常に困難である。

本論文では、それらの問題を回避するため、時定数の対数に対して勾配法を適用する手法を提案する。そして、時定数の学習の有用性を示すわかりやすい例として、ネットワークが適切な振動パターンを生成するように時定数を学習させることに成功した結果を示す。

2 研究方法

2.1 ネットワーク

CTRNN の各ニューロンでは、内部状態を減衰させて、その分入力を取り込む。したがってニューロンの内部状態は次式のような微分方程式により更新される。

$$\tau_j \dot{u}_{j,t} = -u_{j,t} + \sum_{i=1}^N w_{j,i} x_{i,t} \quad (1)$$

ここで τ_j は j 番目のニューロンの時定数、 $u_{j,t}$ は時刻 t における内部状態、 $w_{j,i}$ は i 番目から j 番目のニューロンへの結合重み値、 $x_{j,t}$ は j 番目のニューロンの出力である。計算機シミュレーションでは、このような振る舞いを有限差分近似法により次のように計算する。

$$u_{j,t} = \left(1 - \frac{\Delta t}{\tau_j}\right) u_{j,t-\Delta t} + \frac{\Delta t}{\tau_j} \sum_{i=1}^N w_{j,i} x_{i,t-\Delta t} \quad (2)$$

ここで $\Delta t = 0.01$ はシミュレーション時間間隔である。ニューロンの出力は \tanh 関数を用いて計算する。時定数 τ_j はネットワークダイナミクスの時間スケールを決定する。ニューロンの τ_j が大きい場合、 $\frac{\Delta t}{\tau_j}$ が小さくなるため、ニューロンの内部状態はゆっくりと変化し、以前の内部状態が次の内部状態へと大きく影響を与える。一方 τ_j が小さい場合、 $\frac{\Delta t}{\tau_j}$ が大きくなるため内部状態はゆっくりと変化し、現在の入力が入力状態へと大きく影響を与えるようになる。 $\frac{\Delta t}{\tau_j} = 1$ であると、この式は離散時間ニューロンと等価になる。離散時間ニューロンの場合は Δt の値によって、時定数 τ 、すなわち時間スケールが一意に定まることになる。本論文では、この時定数 τ_j の学習方法を提案する。

2.2 学習方法

本研究では、ネットワークの全ニューロンが直接教師信号を与えられる簡単なケースを設定して実験を行う。時刻 t における誤差関数を次のように定義する。

$$E_t = \sum_{j=1}^N \frac{1}{2} (d_{j,t} - x_{j,t})^2 \quad (3)$$

ここで、 $d_{j,t}$ と $x_{j,t}$ は時刻 t における正しい出力とニューロンの実際の出力である。また、当該時刻 t での重み値の誤差への影響のみに注目する。結合重み値は勾配法を用いて次のように修正される。

$$\Delta w_{ji} = -\eta_w \frac{\partial E_t}{\partial w_{ji}} = -\eta_w \frac{\partial E_t}{\partial x_{j,t}} \frac{dx_{j,t}}{du_{j,t}} \frac{\partial u_{j,t}}{\partial w_{ji}} \quad (4)$$

ここで、 η_w は学習係数と呼ばれる小さな正の定数で、勾配法における一回の修正量を決定する。誤差を減らす方向に重み値を修正するため、この式にはマイナスの符号が付いている。式 (4) は次のように展開することができる。

$$\Delta w_{ji} = \eta_w (d_{j,t} - x_{j,t}) (1 - x_{j,t}^2) \frac{\Delta t}{\tau_j} x_{i,t-\Delta t} \quad (5)$$

次に、同様のやり方によって時定数 τ_j を次式のように学習させることを考える。

$$\Delta \tau_j = -\eta_T \frac{\partial E_t}{\partial \tau_j} = -\eta_T \frac{\partial E_t}{\partial x_{j,t}} \frac{dx_{j,t}}{du_{j,t}} \frac{\partial u_{j,t}}{\partial \tau_j}$$

しかし、実際にこの式によって時定数を学習することは難しい。以下にその理由を説明していく。ある時刻 t での内部状態 $u_{j,t}$ の変化は次式で表される。

$$\Delta u_{j,t} = u_{j,t} - u_{j,t-\Delta t} = \frac{\Delta t}{\tau_j} \left(-u_{j,t-\Delta t} + \sum_{i=1}^N w_{j,i} x_{i,t-\Delta t} \right) \quad (6)$$

この式から、 u_j は τ_j に反比例しており、 τ_j が小さい時わずかな τ_j の変化で u_j が敏感に変化することがわかる。また、 $\frac{\partial u_{j,t}}{\partial \tau_j}$ は次のように計算できる。

$$\begin{aligned} \frac{\partial u_{j,t}}{\partial \tau_j} &= \frac{\partial}{\partial \tau_j} \left\{ \left(1 - \frac{\Delta t}{\tau_j}\right) u_{j,t-\Delta t} + \frac{\Delta t}{\tau_j} \sum_{i=1}^N w_{j,i} x_{i,t-\Delta t} \right\} \\ &= \frac{\Delta t}{\tau_j^2} u_{j,t-\Delta t} + \left(1 - \frac{\Delta t}{\tau_j}\right) \frac{\partial u_{j,t-\Delta t}}{\partial \tau_j} \\ &\quad - \frac{\Delta t}{\tau_j^2} \sum_{i=1}^N w_{j,i} x_{i,t-\Delta t} + \frac{\Delta t}{\tau_j} \sum_{i=1}^N w_{j,i} \frac{\partial x_{i,t-\Delta t}}{\partial \tau_j}, \end{aligned}$$

影響が間接的である第 4 項を無視すると

$$\begin{aligned} &= -\frac{\Delta t}{\tau_j^2} (-u_{j,t-\Delta t} + \sum_{i=1}^N w_{j,i} x_{i,t-\Delta t}) \\ &\quad + \left(1 - \frac{\Delta t}{\tau_j}\right) \frac{\partial u_{j,t-\Delta t}}{\partial \tau_j} \\ &= -\frac{1}{\tau_j} (u_{j,t} - u_{j,t-\Delta t}) + \left(1 - \frac{\Delta t}{\tau_j}\right) \frac{\partial u_{j,t-\Delta t}}{\partial \tau_j} \quad (7) \end{aligned}$$

この式は $P_{j,t} = \frac{\partial u_{j,t}}{\partial \tau_j}$ のように置くことで、再帰的に計算することができる。式 (7) により、 $\Delta \tau_j$ は τ_j^{-2} のオーダーの値になることがわかる。したがって、 τ_j が小さくなると、 $\Delta \tau_j$ は非常に大きくなり、 τ_j はしばしば負の値になってしまう。また τ_j が一度大きくなると $\Delta \tau_j$ が非常に小さくなり、 τ_j はほとんど変化しなくなってしまう。そこで、 τ_j の大きさに合わせて $\Delta \tau_j$ の感度を調整し、さらに τ_j を正の値に保つ方法が必要になってくる。このような要求を満たすため、次のような時定数の対数を導入する。

$$T_j = \log_e \tau_j.$$

T_j を通じて τ_j を更新することで、 τ_j が負の値になることと大きい値からほとんど変化しなくなってしまうことを同時に回避することができる。 $d\tau_j$ と dT_j は次のような関係を持つ。

$$d\tau_j = \tau_j dT_j \quad (8)$$

また、 $\frac{\partial u_j}{\partial T_j}$ と $\frac{\partial u_j}{\partial \tau_j}$ の関係は次のようになる。

$$\frac{\partial u_j}{\partial T_j} = \tau_j \frac{\partial u_j}{\partial \tau_j} \quad (9)$$

式 (8),(9) からわかるように τ_j が小さい時、 $\frac{\partial u_j}{\partial T_j}$ と $\Delta \tau_j$ はともに小さくなり、 τ_j が大きい時、 $\frac{\partial u_j}{\partial T_j}$ と $\Delta \tau_j$ はともに大きくなる。勾配法を用いることにより、 T_j は次のように更新される。

$$\Delta T_j = -\eta_T \frac{\partial E_t}{\partial T_j} = -\eta_T \frac{\partial E_t}{\partial x_{j,t}} \frac{dx_{j,t}}{du_{j,t}} \frac{\partial u_{j,t}}{\partial T_j} \frac{dT_j}{dT_j} \quad (10)$$

τ_j を更新させるため、式 (8) に示す $d\tau_j$ と dT_j の関係を用いて、次のような更新式を得る。

$$\Delta \tau_j = \tau_j \Delta T_j = -\eta_T \tau_j^2 \frac{\partial E_t}{\partial x_{j,t}} \frac{dx_{j,t}}{du_{j,t}} \frac{\partial u_{j,t}}{\partial \tau_j} \quad (11)$$

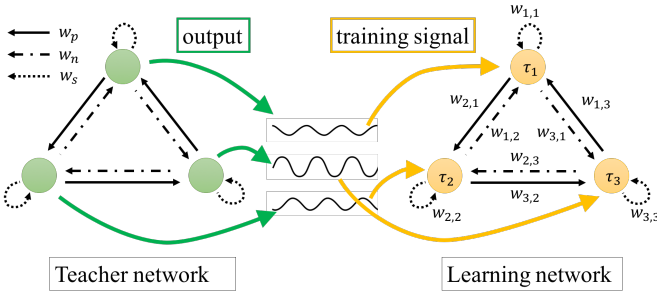


Fig 1: Overview of the oscillator learning task.

ここで、式 (11) は次のように展開することができる。

$$\Delta\tau_j = -\eta_T \tau_j^2 (d_{j,t} - x_{j,t}) (1 - x_{j,t}^2) \frac{\partial u_{j,t}}{\partial \tau_j} \quad (12)$$

このように学習することで、 $\frac{\partial u_{j,t}}{\partial \tau_j}$ は式 (7) から再帰的に計算することができる。式 (12) の中に存在する τ_j^2 が式 (4) の中の $\frac{1}{\tau_j^2}$ の影響を打ち消し、結果として $\Delta\tau_j$ が極端に大きい値か小さい値になることが回避される。

RNN の相互結合を用いたパターン生成の学習においても一つ問題が存在する。もし、各ニューロンの出力が多くの誤差を含んでいた場合、各ニューロンはそれらの誤差を多く含んだフィードバックに基づいて次の時刻の出力を学習してしまうため、結果として学習に失敗してしまう。そこで本研究では、学習中にフィードバックの誤差を小さく保つために、学習中は下記の式を用いて常に教師信号に基づいて内部状態を修正し続けた。

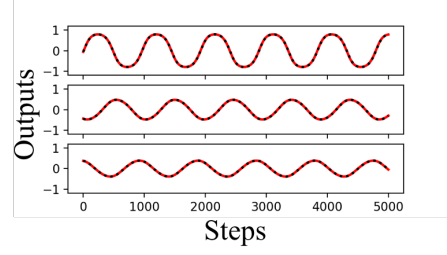
$$\Delta u_{j,t} = \eta_u (d_{j,t} - x_{j,t}) \frac{du_{j,t}}{dx_{j,t}} \quad (13)$$

ここで、 η_u は修正量を調整する正の小さい値 (修正係数) である。同様の手法はリザバコンピューティングによるパターン生成に関する研究でも行われている。Jeager らは学習中に教師信号を出力のフィードバックとして用いる Teacher Forcing と呼ばれる学習法を用いている [4]。Sussillo らは、学習中のすべてのステップで常に誤差が非常に小さくなるように学習を行う FORCE LEARNING を提案している [5]。

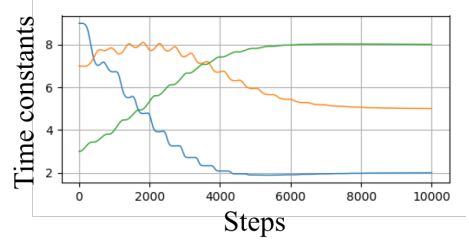
3 結果

3.1 シミュレーションタスク

提案手法をテストするため、Fig.1 に示すような、振動子学習タスクを行った。ここでは、二つの CTRNN を用意する。それぞれは「教師ネットワーク」と「学習ネットワーク」という $N = 3$ つのニューロンを持つネットワークである。教師ネットワークは自律発振するように重み値と時定数を設定する。学習ネットワークには教師ネットワークの出力を教師信号として重み値や時定数の学習を行わせ、学習中のそれらの値の変化の様子や学習の成否を観察する。教師ネットワークに自律発振をさせるため、Fig.1 に示すネットワークの反時計回りの向きへの信号の重み値 w_p とセルフフィードバックの重み値 w_s を正の値に、時計回りの向きへの信号の重み値 w_n を



(a) Outputs (test phase)



(b) Time constants (training phase)

Fig 2: The results in task 1. (a) shows the outputs of the learning network (red line) and the desired output generated by teacher network (dotted line) during test phase. (b) shows the time constants of the learning network during training.

負の値に設定し、内部状態の初期値を一つは 1.0、その他のニューロンは 0 に設定する。このように設定するとニューロンの活性は反時計回りにシフトしていき、ネットワークは自律発振を始める。学習は二つの設定で行う。(Task1) では、二つのネットワークの重み値は同じにして時定数のみを学習させる。(Task2) では、重み値と時定数の両方を違う値に初期設定して学習を行い、重み値のみ学習した場合と両方とも学習した場合を比較する。

3.2 学習結果

教師ネットワークの出力を教師信号として、学習ネットワークは 10000step の間学習を行い、5000step の間テストを行った。学習係数と修正係数はそれぞれ $\eta_w = 20$, $\eta_T = 0.01$, $\eta_u = 0.01$ のように設定した。教師ネットワークの時定数は $\tau_1 = 2.0, \tau_2 = 5.0, \tau_3 = 8.0$ とし、重み値は $w_s = 2.0, w_p = 3.0, w_n = -4.0$ とした。学習ネットワークの初期時定数は $\tau_1 = 9.0, \tau_2 = 7.0, \tau_3 = 3.0$ とし、初期重み値は、Task1 では $w_s = 2.0, w_p = 3.0, w_n = -4.0$ とし、Task2 では全て 0 とした。

Fig.2 に Task1 の学習結果を示す。時定数の学習結果から、学習により学習ネットワークの時定数が教師ネットワークの値と同じ値に収束し、テスト期間の出力は教師信号と一致していることがわかる。時定数は教師信号に基づいて修正され、3 つのニューロンの教師信号の周期は同じであるにも関わらずそれぞれの大小関係を入れ替えて教師ネットワークと完全に一致させることができた。

Fig.3 に Task2 の学習結果を示す。Fig.3(a) は重み値の学習だけを行った場合の出力結果であり、(b)(c)(d) は重み値と時定数の両方の学習を行った時の結果である。重み値の学習だけでは失敗し、時定数の学習を導入した場

合のみ成功していることから、提案手法により時定数の学習を行うことで、学習が時間軸方向に拡張され、学習パフォーマンスの向上が期待される結果が示された。また、(c)(d)に示す学習結果より、重み値と時定数は教師ネットワークと異なる値へと収束しているが、テスト期間の出力は教師信号と一致し、学習に成功していることがわかる。この結果から、正しい出力を生成する重み値と時定数のペアは複数存在することが示唆される。

4 結論

本論文では、すべてのニューロンに教師信号が与えられている時にその時定数を更新させる学習則を提案した。提案手法を用いて一様な周期で自律発振する教師ネットワークと同じ波形を再現させるタスクを行なった。重み値を最適値に固定したまま、時定数のみ学習させたところ、学習によって各ニューロンの時定数は最適な値にそれぞれ収束することを確認できた。重み値と時定数の両方を最適値と異なる値に初期化したところ、重み値の学習だけでは教師信号と同じ波形を再現できなかったが、時定数を学習させることで再現できた。このことから時定数を学習させることが学習の自由度を拡張し、パフォーマンスを向上させることにつながる可能性が示唆された。今後の課題として、多層のCTRNNを学習させるため今回の提案手法を Back Propagation Through Time ができるように拡張すること、より複雑なパターンやダイナミクスを生成したり入力処理する必要のあるようなさらに難しいタスクへの適用などがある。

謝辞

本研究は JSPS 科研費 JP15K00360, JP18H00543 の助成を受けた。

参考文献

- [1] Y.LeCun, Y.Bengio, G.Hinton : Deep learning. Nature 521, 436-444 (2015)
- [2] Y.Yamashita and J.Tani: Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment. PLoS Computational Biology Vol.4, No11. (2008)
- [3] M.C.Mozer: Induction of Multiscale Temporal Structure. Advances in Neural Information Processing Systems 4: 27582. (1992)
- [4] H.Jaeger: Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. Technical Report GMD Report 159, German National Research Center for Information Technology. (2002)
- [5] D.Sussillo, L.F.Abbott: Generating coherent patterns of activity from chaotic neural networks. Neuron article, Vol.63, No.4, pp.544-557(2009)

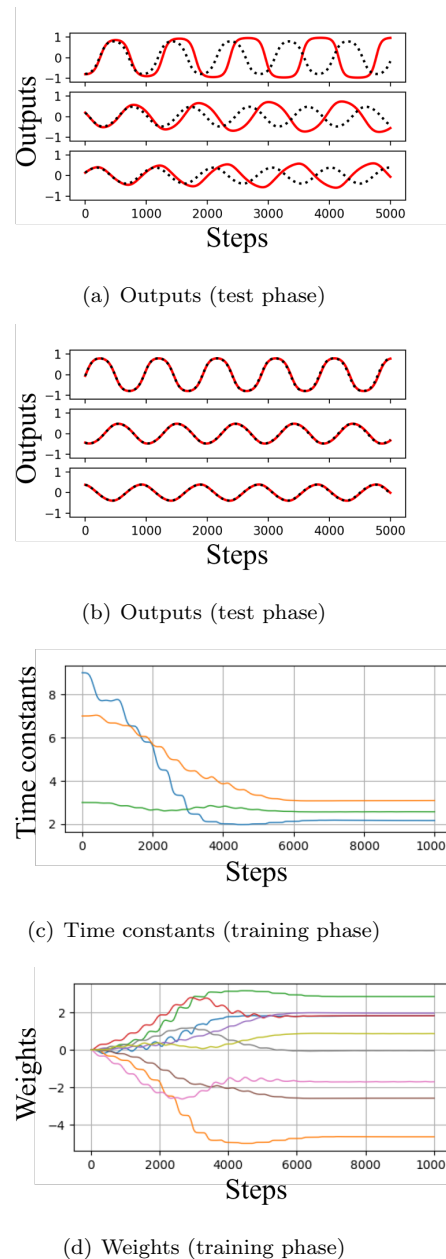


Fig 3: The results in task 2. (a), (d) shows the outputs of the learning network (red line) and the desired output generated by teacher network (dotted line) during test phase. Both of the connection weights and time constants are trained in (a), while the connection weights are only trained in (d). (b), (c) show the time constants and the connection weights of the learning network during training phase respectively.