カオスニューラルネットを用いた記憶問題の学習における不応性導入の影響

大分大学　○黒崎耕平　柴田克成

Effect of Introduction of Refractoriness on Supervised Learning of a Memory Task using a Chaos Neural Network

Kohei Kurosaki and Katsunari Shibata, Oita University

Abstract: We think that "inspiration" or "discovery" is essential property to realize thinking, and emerges as chaotic itinerary between non-fixed point flow-type attractors in which states transition autonomously and rationally. The Chaotic itinerancy can be observed when fixed-point attractors are embedded in a chaotic neural network in which refractoriness is introduced in each neuron, and the way how the network produces chaotic dynamics must deeply influences the maintenance of chaoticity. In this paper, we observed that the difference in Lyapunov exponent during supervised learning depending on whether refractoriness was introduced in each neuron or not. The results show that when refractoriness is introduced, chaos after learning is stronger in a wider range of initial weight scale. When the initial strength of chaos is the same, the hidden outputs take values in unsaturated zone more often due to the smaller weights, and that could lead to the faster learning.

1　まえがき

本研究室では、人間による脳の理解に基づいて人工知能を設計するのではなく、ニューラルネット (NN) と強化学習の組み合わせによって、脳が持つさまざまな働きを人工知能に自律的に獲得させる、という研究を行っている [1]。これまで、NN ＋強化学習によって簡単な記憶や予測の機能の創発が確認されているが [2][3]、高次機能である思考に相当する機能の創発は未だ確認できていない。

また本研究室では、このような機能は合理的な状態遷移を伴うアトラクタ（以下、フロー型アトラクタと呼ぶ）によるものと考えており、その創発のためには、NN へのカオスの導入が有効であると考えている [1]。その理由は大きく 3 つある。1 つ目は「内部ダイナミクスによる探索が可能となるから」である。カオスダイナミクスはそれ自身が不規則な状態遷移をしており、少しでも状態が異なれば、カオスが持つ初期値鋭敏性により全く違う状態へと変化するため、探索が可能になる。2 つ目は「学習が容易になる可能性があるから」である。カオスを導入しないリカレント NN では状態遷移を「作る」必要があるが、カオス NN では、自身が予め持つ不規則な状態遷移を合理的なものに「変える」だけでよいので、学習が容易になると考えている。

そして 3 つ目は「あるアトラクタへの収束後でも別のアトラクタへ遷移できる可能性があるから」である。本研究室では、「思考」に相当する機能の創発には、合理的なフロー型アトラクタのダイナミクスだけでなく、「ひらめき」や「発見」のような創造性を実現するために、形成されたこれらの複数のフロー型アトラクタの間を不規則に遷移することが必要であると考えている。カオス NN に状態遷移を伴わない固定点アトラクタを埋め込むと、カオス的遍歴と呼ばれるアトラクタ間の不規則な遷移が起きることが知られている [4]。そこで、フロー型アトラクタ間でもカオス的遍歴が起これば、「ひらめき」や「発見」に相当する機能の創発につながると期待している。本研究では、この 3 つ目に注目する。

しかしながら現段階では、カオス NN を用いた強化学習アルゴリズムは確立されておらず、加えて、この強化学習アルゴリズムによる、フロー型アトラクタの形成も観察できていない。そこで本研究では、まずはカオス NN に教師あり学習によって記憶が必要なタスクを学習させ、そのときの NN の挙動について詳しく調べることとした。

カオス NN を構築するには一般的に、相互に結合されたリカレント NN を用いるが、その結合重み値を大きな乱数にする方法と、それに加えてさらに、各ニューロンに不応性を導入する方法がある。前者の場合、学習による固定点アトラクタの形成は確認できたが、カオス的遍歴は観察できず、カオス性の低下が見られた [5]。また、フロー型アトラクタとして振動子を埋め込んだが、やはりカオス的遍歴は観察されなかった [6]。一方、ニューロンに不応性を導入した場合、前述のように埋め込んだ固定点アトラクタ間のカオス的遍歴が観察される。これらのことから、相互結合重み値を大きくするだけでは、学習によってその重み値自身が変化するため、カオス性を維持することは難しいが、不応性を各ニューロンに導入することで、カオス性の維持が容易になると考えた。

そこで本研究では、不応性を導入したニューロンによってカオスダイナミクスを発現させるカオス NN に、記憶が必要で簡単なタスクとして、入出力に時間差をもうけた EX-OR を教師あり学習させ、不応性の有無による学習前後の重み値とカオス性の関係の違い、および、両者の学習時の違いを観察した。

2　カオスニューラルネットワークとその学習

本研究では、不応性を導入した動的なニューロンを相互結合した中間層と静的なニューロンによる出力層によっ

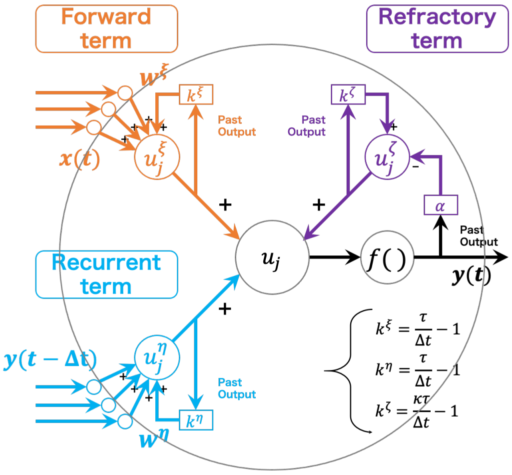


Fig. 1: A chaotic neuron model with refractoriness.

て構成される 3 層のカオス NN を使用した。不応性を導入したニューロンモデルを Fig. 1 に示す。

このニューロンモデルは、時刻 t における、m 個の外部入力 $\boldsymbol{x}_t = (x_{1,t},\cdots,x_{m,t})^T$ を受け取る ξ 項、同層の n 個のニューロンからのリカレント入力 $\boldsymbol{y}_{t-\Delta t} = (y_{1,t-\Delta t},\cdots,y_{n,t-\Delta t})^T$ を受け取る η 項、実際の生物ニューロンの性質をもとに導入された不応性を担う ζ 項により構成される。

各項の内部状態 $u^{\beta}_{j,t}$ $(\beta = \xi,\eta,\zeta)$ は式 (1)〜式 (3) によって求められる。ここで、τ は時定数、$w^{\gamma}_{j,i}$ $(\gamma = \xi,\eta)$ は各項の i 番目の入力に対する重み値、α は不応性のスケーリングパラメータ、κ は不応期のスケーリングパラメータである。Δt は本論文では 1 とした。

$$u^{\xi}_{j,t} = \left(1-\frac{\Delta t}{\tau}\right)u^{\xi}_{j,t-\Delta t} + \frac{\Delta t}{\tau}\sum_{i=1}^{N} w^{\xi}_{j,i}\cdot x_{i,t} \tag{1}$$

$$u^{\eta}_{j,t} = \left(1-\frac{\Delta t}{\tau}\right)u^{\eta}_{j,t-\Delta t} + \frac{\Delta t}{\tau}\sum_{i=1}^{N} w^{\eta}_{j,i}\cdot y_{i,t-\Delta t} \tag{2}$$

$$u^{\zeta}_{j,t} = \left(1-\frac{\Delta t}{\kappa\tau}\right)u^{\zeta}_{j,t-\Delta t} - \frac{\Delta t}{\kappa\tau}\,\alpha\; y_{j,t-\Delta t} \tag{3}$$

ニューロンの内部状態 $u_{j,t}$ は式 (4) によって各項の内部状態 $u^{\beta}_{j,t}$ の和として求められ、出力 $y_{j,t}$ は式 (5) によって非線形変換を行うことで求められる。ここで、g は活性化関数のゲインである。

$$u_{j,t} = u^{\xi}_{j,t} + u^{\eta}_{j,t} + u^{\zeta}_{j,t} \tag{4}$$

$$y_{j,t} = f(g\cdot u_{j,t}) = \frac{1}{1+\exp(-g\cdot u_{j,t})} \tag{5}$$

カオス NN の学習には、不応性項の影響を除いた、簡易的な BPTT(Back Propagation Through Time) 法を用いた。式 (6) のように、教師信号 T と出力層の出力 $o^{(\mathrm{o})}_{j,t}$ で二乗誤差 E が定義される。ここで、$N^{(\mathrm{o})}$ は出力層のニューロン数である。

$$E_{t'} = \sum_{j=1}^{N^{(\mathrm{o})}} \frac{1}{2}(T_{j,t} - o^{(\mathrm{o})}_{j,t})^2 \tag{6}$$

出力層の誤差信号 $\delta^{(\mathrm{o})}_{j,t}$ は式 (7) によって、重み値の更新量 $\Delta w^{(\mathrm{o})}_{j,i}$ は式 (8) によって求められる。ここで、η は学習係数である。

$$\delta^{(\mathrm{o})}_{j,t} = (T_{j,t} - o^{(\mathrm{o})}_{j,t})\cdot f'(u^{(\mathrm{o})}_{j,t}) \tag{7}$$

$$\Delta w^{(\mathrm{o})}_{j,i} = \eta\cdot\delta^{(\mathrm{o})}_{j,t}\cdot o^{(\mathrm{h})}_{i,t} \tag{8}$$

教師信号が与えられる時刻 t' における中間層の誤差信号 $\delta^{(\mathrm{h})}_{j,t'}$ は式 (9) によって、それ以外の時刻 t における中間層の誤差信号 $\delta^{(\mathrm{h})}_{j,t}$ は式 (10) によって求められる。ここで、$N^{(\mathrm{h})}$ は中間層のニューロン数である。

$$\delta^{(\mathrm{h})}_{j,t'} = \left(\sum_{k=1}^{N^{(\mathrm{o})}} \delta^{(\mathrm{o})}_{k,t'}\cdot w^{(\mathrm{o})}_{k,j}\right)\cdot f'(u^{(\mathrm{h})}_{j,t'}) \tag{9}$$

$$\delta^{(\mathrm{h})}_{j,t} = \frac{\Delta t}{\tau}\left(\sum_{k=1}^{N^{(\mathrm{h})}} \delta^{(\mathrm{h})}_{k,t+\Delta t}\cdot w^{\eta}_{k,j}\right)\cdot f'(u^{(\mathrm{h})}_{j,t}) + \left(1-\frac{\Delta t}{\tau}\right)\delta^{(\mathrm{h})}_{j,t+\Delta t} \tag{10}$$

入力層との結合およびフィードバック部の結合の重み値の更新量 $\Delta w^{\xi}_{j,i,t}$、$\Delta w^{\eta}_{j,i,t}$ はそれぞれ式 (11)、(12) によって求められる。η^{ξ},η^{η} は、それぞれの学習係数である。

$$\Delta w^{\xi}_{j,i,t} = \eta^{\xi}\cdot\delta^{(\mathrm{h})}_{j,t}\cdot\frac{\Delta t}{\tau}\cdot o^{(\mathrm{in})}_{i,t} \tag{11}$$

$$\Delta w^{\eta}_{j,i,t} = \eta^{\eta}\cdot\delta^{(\mathrm{h})}_{j,t}\cdot\frac{\Delta t}{\tau}\cdot o^{(\mathrm{h})}_{i,t-\Delta t} \tag{12}$$

3 学習シミュレーション

3.1 学習タスクとネットワークの構成パラメータ

学習タスクは入出力に時間差をもうけた EX-OR とした。学習を簡単にするため、入力は $-1,1$ とし、アトラクタの形成を促進するため、個々の入力に $[-0.2,\ 0.2]$ の一様乱数によるノイズを付与した。入力パターンと教師信号を Table 1 に、入力信号および教師信号の入力タイミングを Fig. 2 に示す。

また、タスクの学習中はカオス性の強さを示す指標であるリアプノフ指数を観察した。リアプノフ指数が正となればカオス性があることを示す。本シミュレーションでは、入力が入った時刻 $t=0$ における中間層ニューロンの内部状態 $\boldsymbol{u_0}$ に、0.001 の大きさに正規化した乱数を加えて、$T=20$ ステップネットワークを回し、乱数を加えたものと加えなかったものの距離 d_t からリアプノフ指数を

Table 1: Input Pattern and Teaching Signals.

Input Pattern	Input Signals		Teaching Signal
	input 1	input 2	
0	-1	-1	0.1
1	-1	1	0.9
2	1	1	0.1
3	1	-1	0.9

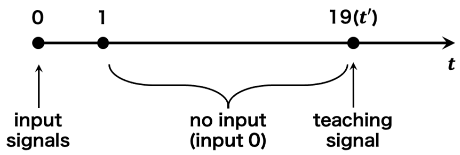


Fig. 2: Timing of Input and Output.

求めた。これを乱数系列 $P = 10$ 個において、式 (13) のように計算した。

$$\lambda \;=\; \frac{1}{T \cdot P}\sum_{p=1}^{P}\sum_{t=1}^{T}\ln\frac{d_{p,t+1}}{d_{p,t}} \tag{13}$$

本シミュレーションのパラメータ設定を Table 2 に示す。中間層の初期のフィードバック重み値 $\boldsymbol{W}^{\eta}$ は、スペクトル半径が Scale 倍となるように乱数で設定した。

Table 2: Parameters.

Name		Value
Episode		3000
Gain of Sigmoid Function		2
τ		1.25
Number of Neurons in Hidden Layer		100
Learning Rate	Output	0.1
	Hidden(ξ)	0.3
	Hidden(η)	0.01
Initial Weight	Output	$[-0.3, 0.3]$
	Hidden(ξ)	$[-0.3, 0.3]$
	Hidden(η)	Varied Scale
Parameters for Refractoriness	α	3
	κ	8

3.2 シミュレーション結果

まず、中間層のフィードバック部の重み値スケールを 1.0〜20.0 の範囲で変化させたときの、不応性の有無によるリアプノフ指数の変化について、Fig. 3 は学習前、Fig. 4 は学習後の結果を示す。それぞれ乱数系列 10 個の平均をとった結果である。

Fig. 3 より、同じ重み値スケールにおいて、全体的に不応性ありの方が学習前のリアプノフ指数が高い値を示し

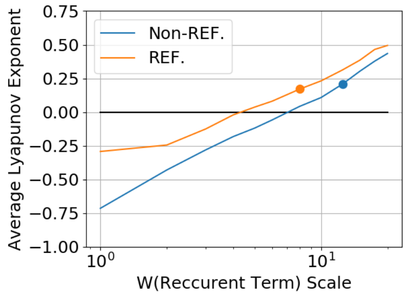


Fig. 3: Comparison of Lyapunov Exponent (before learning). The dots indicate the scales used in Fig. 5-8.

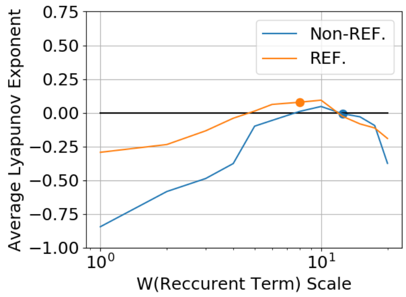


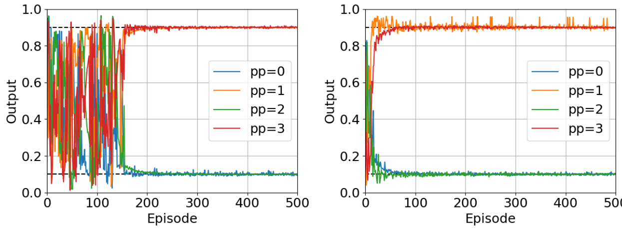
Fig. 4: Comparison of Lyapunov Exponent (after learning).

ていることがわかる。また、不応性ありの場合は、重み値スケールが 5.0 あたりでカオス性を示し始めていることがわかる。重み値のスケール 5.0 と、ゲイン 2 のシグモイド関数の微分値がとりうる最大値 0.5 をかけると 2.5 と大きいが、動的ニューロンを用いていること、微分値が実際には 0.5 より小さいためと考えられる。

また、Fig. 4 を見ると、重み値スケールが大きいときは学習後のリアプノフ指数がいずれも大きく下がっているが、カオス性を有するリアプノフ指数が 0 より大きいところを見ると、不応性ありの場合の方がリアプノフ指数が大きく、また、リアプノフ指数が 0 を超えている範囲も不応性がある時の方が広いことがわかる。

次に、学習前のリアプノフ指数を揃えた状態での学習中の様子を比較するために，不応性ありで重み値スケールを 8.0、不応性なしで重み値スケールを 12.5 とした場合の学習の様子の例を示す。Fig. 5 に学習による出力の変化を示す。不応性ありの場合の方が学習が早く，不応性がない場合は学習が不安定で収束が遅くなっていることがわかる。このことは、他の乱数系列の場合にも同様の傾向が見られた。学習中のリアプノフ指数の比較結果を Fig. 6 に示す。学習初期のリアプノフ指数はほぼ等しいものの、不応性なしの場合は，学習するにしたがって、リアプノフ指数が大きく減少しているのに対し、不応性ありの場合は大きな減少は見られない。

さらに、中間層ニューロンの出力の時間変化を、パターン 0 の場合を例に、不応性ありの場合を Fig. 7 に、不応性なしの場合を Fig. 8 に、それぞれ学習前と学習後の場



(a) without refractoriness (Non-REF: $\boldsymbol{W}^{\eta}$ Scale=12.5)

(b) with refractoriness (REF: $\boldsymbol{W}^{\eta}$ Scale=8.0)

Fig. 5: Comparison of output change during learning.

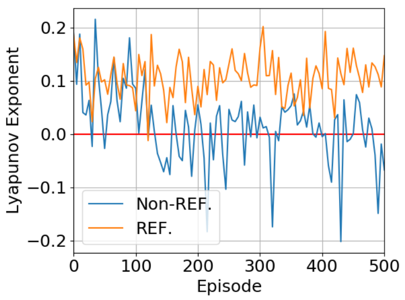


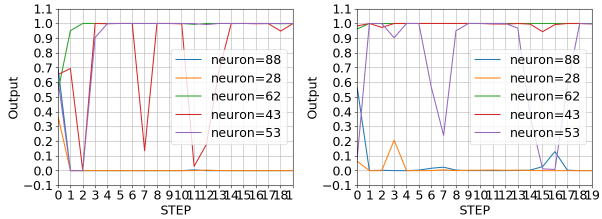
Fig. 6: Comparison of Lyapunov Exponent during learning.

合を並べて示す。中間層ニューロンは、学習後の中間層→出力層の結合重み値が大きい順に 5 つをピックアップした。いずれの場合も、学習前と学習後を比較しても、あまり大きな違いは見られない。また、不応性なしの場合は、シグモイド関数の飽和領域である 0 や 1 に近い値を出力する傾向がある一方、不応性ありの場合は、中間的な値を出力する傾向があることがわかる。これは、同じカオスの強さを実現するためには、不応性なしの場合にはより大きな重み値に設定する必要があったためと考えられる。また、中間層ニューロンの値が中間的な値を取らずに 0 や 1 に大きく変動するために、Fig. 5 のように学習が不安定になって遅くなったのではないかと考えられる。

4　あとがき

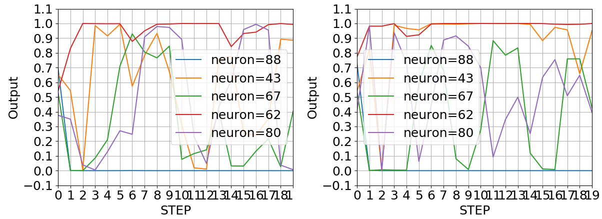
不応性を導入したニューロンを導入した場合としない場合のカオス NN を用いて、記憶を必要とするタスクの教師あり学習を行なった。その結果、同じカオスの強さを実現するためには、不応性ありの方が重み値スケールが小さくてすみ、それが、学習の安定につながると考えられる。また、学習後は、不応性ありの場合の方がより大きなリアプノフ指数を実現でき、さらに、リアプノフ指数が 0 より大きくなる初期重み値の範囲も広かった。このことから、学習後のカオス性を維持するためには不応性を導入したニューロンを用いる方が良いと考えられる。

今後は、本来の目的である、学習によって形成されたアトラクタ間のカオス的遍歴が見られるかどうかを確認していきたい。



(a) Before learning　　(b) After learning

Fig. 7: Comparison of hidden outputs. (Non-REF: $\boldsymbol{W}^{\eta}$ Scale=12.5, pattern=0)



(a) Before learning　　(b) After learning

Fig. 8: Comparison of hidden outputs. (REF: $\boldsymbol{W}^{\eta}$ Scale=8.0, pattern=0)

謝辞

本研究は JSPS 科研費 (15K00360, 20K11993) および栢森情報科学技術振興財団研究助成金の補助を受けたものである。ここに謝意を表する。

参考文献

[1] 柴田克成: End-to-End 強化学習による知能創発と「思考」創発へ向けた新しい強化学習, 第 62 回システム制御情報学会研究発表講演会講演論文集 (2018)

[2] H. Utsunomiya & K. Shibata: Contextual Behavior and Internal Representations Acquired by Reinforcement Learning with a Recurrent Neural Network in a Continuous State and Action Space Task, ICONIP 08, Vol. 5507, pp. 970-978 (2009)

[3] K. Shibata & K. Goto: Emergence of Flexible Prediction-Based Discrete Decision Making and Continuous Motion Generation through Actor-Q-Learning, Proc. of ICDL-Epirob 2013 (2013)

[4] 小室元政: 大域結合写像におけるカオス的遍歴の発生機構, 数理解析研究所講究録 1118 巻, pp.96-114(1999)

[5] 黒崎耕平: カオスニューラルネットワークの教師あり学習における固定点アトラクタの発現, 2018 年度大分大学工学部電気電子工学科卒業論文 (2019)

[6] 松矢直樹: カオスニューラルネットにおける埋め込み振動子間の遷移の観察, 2019 年度大分大学工学部電気電子工学科卒業論文 (2020)