カオスフラクタル　最終課題レポート

2023/8/6

みかさん

概要

非線形ダイナミクスの応用例ということだったのですが、正直アートぐらいしか思い浮かびませんでした。（とりあえず）レザバーコンピューティングでカオスに関係しているらしい研究を見つけたので翻訳した内容を頑張って書きます

1　リザバーコンピューティングとは

- リザバーコンピューティングは時系列予測をおこなう機械学習手法の一つ。何らかの潜在的な情報を持ったレザバーに入力し、最終層の重みのみを学習することで、明示的なアーキテクチャやモジュールに頼らずモデルを設計することができる。
- 通常の RNN におけるリカレント層は、ほかの層よりノード数が多い。
  また、全結合層を利用した場合の 1 エポック当たりの時間計算量は $O(TN_x^2)$ であり、空間計算量は $O(TN_x)$ である.
  RNN は勾配消失・爆発が起こりやすいことで有名. 一般的には LSTM、GRU を用いる。
  - T：系列長
  - N_x：入力ベクトルの大きさ
  - 時間計算量とは：アルゴリズムを一通り計算するのに必要な計算の回数
  - 空間計算量とは：アルゴリズムを一通り計算するのに必要なメモリの大きさ
- **ESN(Echo State Network)**：2001 年に Jaeger によって発見されたリザバーコンピューティングの手法で、一般的な RNN と同様の構造を持つが出力層につながる出力結合重みのみを学習する。学習は誤差逆伝播法を用いる。
  ESN は一定の条件を満たす任意の離散時間フィルタを任意の精度で近似できるという普遍的な離散時間フィルタ近似能力を持つことが知られている。ESN はいくつかのモデルが提案されている。
  -
  -
  -
- **LI モデル**：神経発火モデルの一つで、これをリザバーのノードに用いることでモデルの状態変動の速さを制御する。

$$x_{n+1} = (1-\alpha)x_n + \alpha f(W^{in}u(n+1) + Wx_n$$

$$y_{n+1} = W^{out}x_{n+1}$$

$$\alpha \in (0, 1]$$

- **ESP(Echo State Property)**：リザバー層に使用する重みは、リザバー内の状態ベクトル x_n がその初期値 x_0 と入力ベクトル $u_0, u_1, u_2, ...$ から一意に決まり、時間が経つと入力ベクトルのみに依存するようになることを保証する必要がある。

このことは、以下の式によって示される。基本モデルにおけるリザバーの状態更新式は、ある状態遷移関数 $F : R_u^N \times R_x^N \Rightarrow R_x^N$ を用いて

$$x(n+1) = F(u(n+1), x(n))(n = 0, 1, 2, ...)$$

$$x_0, F(u(n), \tilde{F}(s_{n-1})(u), x_0)$$

このとき ESP を満たすとは、以下の式を満たすということ。

$$lim_{n\to\infty}||\tilde{F}(s_n(u), x_0) - \tilde{F}(s_n(u), z_0)||_2 = 0$$

この式は異なる初期値から出発したリザバーの挙動が徐々に一致するということを示している。

-

2 カオス的遍歴とは

- **カオス的遍歴 (CI)**：1990 年ごろに金子（大域的結合カオス系:GCM)、池田（光学乱流)、津田（神経科学）によってそれぞれ別々に発見された。高次元の力学系においていくつかのほぼ安定な状態が存在している。低次元のアトラクタに似た安定した状態の一つにしばらくの間滞在した後、別の安定した状態に移り、いくつかの秩序状態のカオスを経て遍歴する現象である.
  - **大域結合写像 (Globally Coupled Model:GCM)**：パラメータ $a(1 \leq a \leq 2)$ を持つ一次元のロジスティック写像はカオスを示すことが知られている。(図 1)

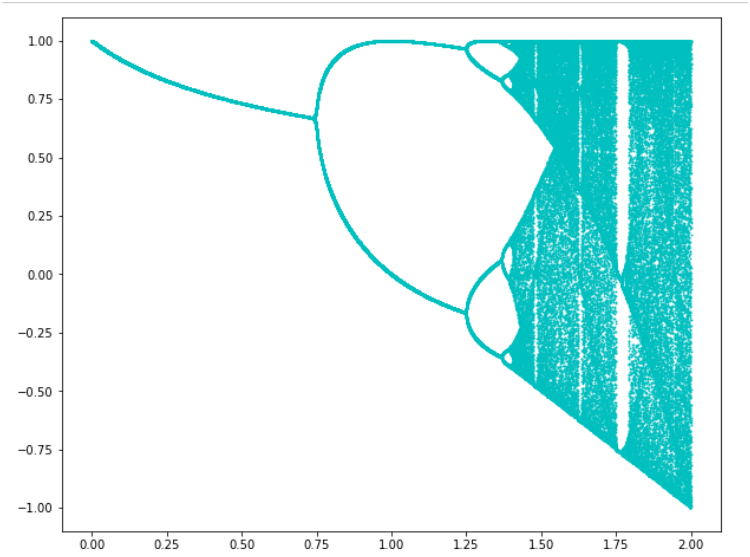


図 1　ロジスティックマップ

このようなカオス的な挙動を示す関数 F を用いて、

$$x^i_{n+1} = (1-\epsilon)f(x^i_n) + \frac{\epsilon}{N}\sum_{n=1}^{N} f(x^j_n)(i = 1, 2, .., N)$$

という式に従う結合振動子の動きを写像する。例えば、N=3 の場合は

$$x^1_{n+1} = (1-\epsilon)f(x^1_n) + \frac{\epsilon(f(x^2_n) + f(x^3_n))}{3}$$

$$x^2_{n+1} = (1-\epsilon)f(x^2_n) + \frac{\epsilon(f(x^1_n) + f(x^3_n))}{3}$$

$$x^3_{n+1} = (1-\epsilon)f(x^3_n) + \frac{\epsilon(f(x^1_n) + f(x^2_n))}{3}$$

(x^i_n ： n 回目の写像 x の i 番目の要素)

– **GCM の挙動**：例として挙げた写像をクラスタ数の観点で観測すると、以下のような結果が得られる。金子は N=200, a を 0.01 刻み、ϵ を 0.02 刻みで変えて調べた。（初期値はランダム）すると、以下のような層が観察された。

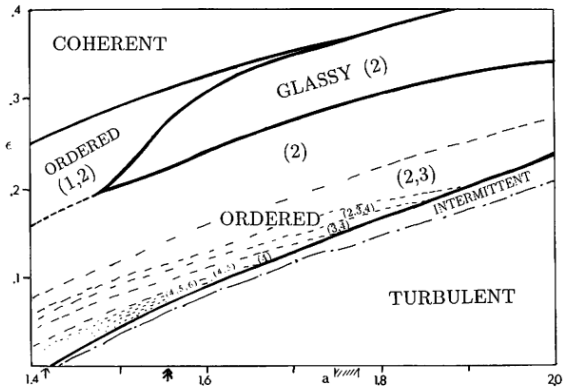


Fig. 3. Rough phase diagram: Phases are determined by $Q(k)$, calculated from 500 randomly chosen initial conditions and $N = 200$. The parameters are changed from $a = 1.4$ to 2.0 by 0.01 and $\epsilon = 0.02$ to 0.4 by 0.02. Numbers such as (1, 2, 3) represent dominant cluster numbers (with basin volume ratio more than 10%). The single arrow at the bottom line shows accumulation of period-doubling bifurcations, while the double arrow denotes the band merging point for the single logistic map. The region with oblique lines correspond to the period-3 window in the logistic map.

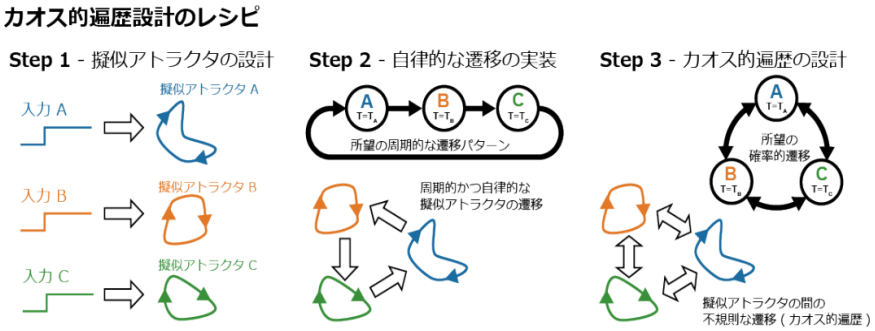
図 2　http://chaos.c.u-tokyo.ac.jp/papers/gcm/phys90.pdf, [3]

* **コヒーレント層**：クラスタは 1 個で、a（非線形のパラメータ）の値が小さく、ϵ（結合の強さを示すパラメータ）が大きい場合に発生する。
* **非同期層**：クラスタ数は N 個、つまり N 個の変数は全てばらけて変化する。a が大きく ϵ が小さい場合に起こる。
* **秩序層**：いくつかのクラスタにわかれて動くが、クラスタ数は N より非常に小さい。構成比やどのような状態に落ち着くかは初期値によって変化する。クラスタの組み換えは起こらない。
* **部分秩序層**：クラスタの組み換えが完結的に起こる。部分秩序相は 2 つの領域に分かれており, **秩序相と非同期相との間に位置する領域**と、**コヒーレント相と秩序相との間に位置する領域**が存在している。

3　高次元カオスを活用したカオス的遍歴の設計手法の提案

このセクションは参考文献 [4], [5] の内容をまとめ？？たもの。（書きかけ。）

- **研究の背景**：
  動物の脳の活動はカオス的遍歴を示すことが指摘されており、これは動物の自発的な行動生成に重要な役割を果たすと考えられている。この方法で学習が進められれば、従来のように状況判断と行動を一対一で設計する必要がない。したがってニューロロボティクスではカオス的遍歴を上手く設計して動物の自発的行動を実装するための研究がおこなわれてきた。しかし、一般に非線形力学系を制御することは困難である。この研究では、リザバーコンピューティングの内部状態の時間発展の複雑さを利用して自発的な行動の切り替えを行うカオス的遍歴を設計する手法を紹介している。
- **内容**：
  高次元のカオス力学系における擬似アトラクタの軌道と遷移則を設計するアルゴリズムを提案
- **提案手法**：以下の手順からなるバッチ学習を提案する。



(図 1) 学習セットアップ及びカオス的遍歴設計のレシピ

図 3　https://www.u-tokyo.ac.jp/content/400149434.pdf

1. **Designing quasi-atractor（擬似アトラクタの設計）**：
   目的の出力ダイナミクスに対する高い操作性をもつ内部ダイナミクス設計する。ターゲットの擬似アトラクターが埋め込まれた高次元カオスシステムを用意する。高次元カオスシステムとして、RNN の一種である ESN を用いた。この ESN は、階層構造やモジュール（例えば、複数の時間スケール）を含まず、全てのネットワークノードが同じ時間スケールパラメータを共有する
   同時に、システムが、離散入力 (内部パラメータ) の種類に対応する初期カオスシステム（生得的軌道）によって生成される固有の複雑な軌道を再現性よく生成するように、内部パラメータを修正する。また指定された軌道（出力ダイナミクス）を出力するように線形回帰モデル（readout と命名）を訓練する。
   このプロセスはリザーバーコンピューティングの手法を用いる。学習するパラメータが少ないので、誤差逆伝播法より学習は安定しており、計算コストは低い。
2. **Embedding autonomous transitions of symbol（自律的な遷移の実装）**：
   フィードバックループを行う識別機を追加することで、特定のシンボルを自動的に生成する。識別機の学習はリザバー内部のパラメータを固定して行う。これにより、軌道の遷移則を最小限の計算機能力（記憶容量や非線形性）で獲得する。
3. **Embedding stochastic transitions of symbol (カオス的遍歴の設計)**：
   有限状態マシンによって制御されるいくつかの確率的シンボル遷移規則を準備し、システムがシステムのカオス性を利用してこれらの確率的ダイナミクスをシミュレートする

- **アーキテクチャの説明**:
  - 入力 ESN とカオス ESN のダイナミクスは次の方程式によって与えられる。

$$\tau \frac{d\mathbf{x}^{in}}{dt}(t) = -\mathbf{x}^{in}(t) + tanh(g^{in} J^{in} \mathbf{x}^{in}(t) + \mathbf{u}^{in}(s(t)))$$

$$\tau \frac{d\mathbf{x}^{ch}}{dt}(t) = -\mathbf{x}^{ch}(t) + tanh(g^{ch} J^{ch} \mathbf{x}^{ch}(t) + J^{ic} \mathbf{x}^{in}(t))$$

  これをまとめると、次のような式になる。

$$\tau \frac{d\mathbf{x}}{dt}(t) = -\mathbf{x}(t) + tanh(\mathbf{g} \odot J(\mathbf{x}(t)) + \mathbf{u}(s(t)))$$

  ただし、次のように定める。

$$\mathbf{x}(t) := [\mathbf{x}^{in}(t); \mathbf{x}^{ch}(t)]$$

$$\mathbf{g} := [g^{in}, g^{in}, ..., g^{in}, g^{ch}, g^{ch}, ..., g^{ch}]$$

$$J := [J^{in}, 0, J^{ic}, J^{ch}]$$

$$\mathbf{u}(s) := [\mathbf{u}^{in}(s); 0]$$

また出力ダイナミクスは、内部ダイナミクスの線形変換によって計算される。つまり、線形読み出しを行う最終層の w_{out} は次の目的ダイナミクスを予想するように訓練される。

$$w_{out}x(t) = f_{out}(t)$$

ステップ 1 で与えられたシンボリックダイナミクス $s(t)$ はステップ 2 で与えられた閉じた系で自動的に生成される。フィードバックループでは、次の分類器 f_{max} が

$$f_{max}(\mathbf{x}(t)) = argmax_{s\in S}\mathbf{w}_s^T\mathbf{x}(t)$$

w_s は、その要素が自動的に指定されたシンボリックダイナミクス $s(t)$ を真似するように訓練された接続行列を示している。
つまり、擬似アトラクタ、出力ダイナミクス、シンボリックダイナミクスをそれぞれ、RNN 接続行列の J, 線形読み出しの W_{out}, 分類器 W_S のパラメータを調整することで設計する。

– ここでは、二つのリザバーコンピューティング技術（innate training と FORCE）を使用している。
FORCE 学習 :First Order-Reduced and Controlled-Error
力学系のカオス性を利用して、指定されたダイナミクスを系に埋め込む方法。単一フィードバックループを使用した以下のダイナミクスについて考えると、

$$\tau\frac{d\mathbf{x}}{dt}(t) = -\mathbf{x}(t) + tanh(gJ\mathbf{x}(t)) + \mathbf{u}(z(t)))$$
$$z(t) = w^T x(t)$$

u は線形フィードバックベクトル。G は scaling parameter（正則化パラメータか、正規化パラメータか..??）で系全体をカオスにするために 1 より大きい数に設定される。また、w は目標ダイナミクス $f(t)$ をシステムに埋め込むために次の C_{out} をコスト関数としてパラメータを調整する。

$$C_{FORCE} = \|z(t) - f(t)\|^2$$

ブラケットは複数のサンプルと試行の平均値を示している。w は最小二乗法を用いたオンライン学習で学習する。
innate training? : 適切な日本語??
カオス系が、複雑な時空間パターンを持つ軌道を良く再現できるようにする手法。はじめの RNN で生成されたカオスを再現するため、以下のコスト関数を最小化するように ESN の接続行列 J を修正する。

$$C_{innate} := \|\mathbf{x}(t) - \mathbf{x}_{target}(t)\|^2$$

参考文献

[1] 田中 剛平, 中根 了昌, 廣瀬 明 . リザバーコンピューティング ー時系列パターン認識のための高速機械学習の理論とハードウェアー . 森北出版会社 , 2021

[2] Kaneko Kunihiko, Tsuda Ichiro. 2003-08. ”Chaotic Itinerancy”. Chaos, 13(3), 926-936. ”https://eprints.lib.hokudai.ac.jp/dspace/bitstream/2115/8486/1/ci-focus.preface.pdf”

[3] Kunihiko KANEKO. 1998-05-16. ”Clustering coding, switching, hierarchical ordering, and control in a network of chaotic elements”. ”http://chaos.c.u-tokyo.ac.jp/papers/gcm/phys90.pdf”.

[4] 井上 克馬, 中嶋 浩平, 國吉 康夫, ” 高次元カオスを活用したカオス的遍歴の設計手法の提案”, (https://www.u-tokyo.ac.jp/content/400149434.pdf, 2023-08-23 参照)

[5] Katsuma Inoue, Kohei Nakajima, Yasuo Kuniyoshi, 2020, ”Designing spontaneous behavioral switching via chaotic itinerancy. Science Advances”, (https://www.science.org/doi/10.1126/sciadv.abb3989)