

Reservoir Computer による外力付きカオス時系列モデルの予測と 生体リズム研究への応用

発表者：久野証

東京大学工学部計数工学科数理情報工学コース 4 年

December 7, 2023

目次

- ① 背景
- ② 手法
- ③ 結果
- ④ まとめと展望
- ⑤ Appendix
 - Reservoir Computer の原理
 - Reservoir Computer に関する理論
 - 先行研究
 - Hyperparameters の最適化
 - 実験のパラメータ設定

LD Cycle による生体リズムの研究

概日リズムの研究：LD Cycle

- 常態的な時差ぼけや交代制勤務は、生活習慣病のリスクを高める。
- LD Cycle は生体リズムのモデルに用いられ、光と暗闇の情報が外力として系に加わる。
- 外力や各パラメータの設定により周期的・カオス的な振る舞いを持つ。
- 一般にカオスの予測は難しい。
 - ▶ 初期値鋭敏性。

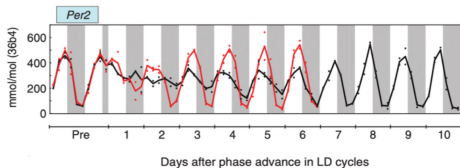


Figure: Day 1 に 8 時間の Jet Lag を受けた時のマウスの Per2 の推移
Image: Fig.2 from [3].

Reservoir Computer を用いたカオス予測 (1/2)

- 右側のアイテム 1
- 右側のアイテム 2

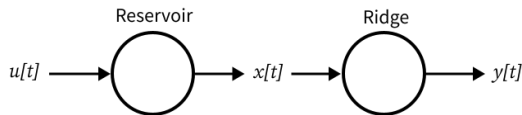
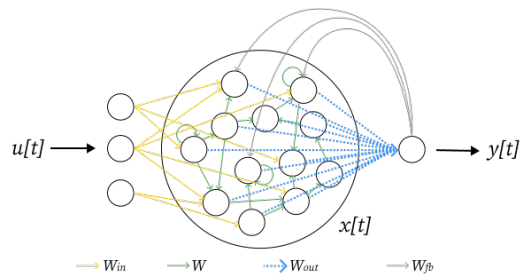


Figure: Reservoirpy での reservoir computer の構造
Image: ReservoirPy, MIT License.

Reservoir Computer を用いたカオス予測 (2/2)

予測の手法

① 予測する系の時系列データを生成

- ▶ 外力付きの Rössler 方程式の系の場合、変数 X_t, Y_t と外力 P_t から成る配列。

② Reservoir の Hyperparameter の最適化

- ▶ train 期間 で学習を行い、test 期間で教師付きの予測を行う。
- ▶ test 期間での予測値と真の値との誤差を目的関数として、最適化を行う。

③ self-evolve 予測を行う。

- ▶ train, warm-up 期間 の後、Reservoir に未来予測をさせる。
- ▶ 各ステップの入力は、Reservoir の一期前の出力に対して、外力の真の値だけ修正した配列。



Reservoirが自身のX、Yの予測値と外力P(t)の真の値を入力に次のステップを予測

Figure: Reservoir Computer を用いたカオス予測の手順

周期外力のある Rössler モデル (1/2)

Rössler 方程式

次の方程式を解くことで時系列データを生成する.

$$\frac{dx}{dt} = -y - z + P(t) \quad (1)$$

$$\frac{dy}{dt} = x + ay \quad (2)$$

$$\frac{dz}{dt} = b + z(x - c) \quad (3)$$

ここで, $P(t) := A \sin(t + \theta_p(t))$ とする.
 ただし, $p \in \{n \in \mathbb{Z} \mid -12 \leq n \leq 12\}$ とし,
 θ_p は 4 日に 1 度外力の位相を $p \cdot 2\pi/24$ だけ早めるような関数.

Rössler Attractor with P(t)

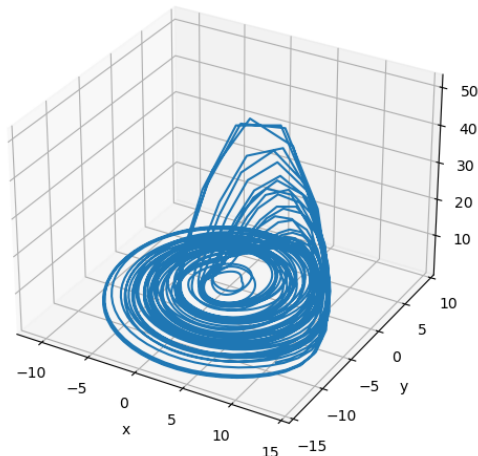


Figure: Rössler モデル

周期外力のある Rössler モデル (2/2)

- Hyperparameter の最適化は位相シフトが 8 時間である系に対して行う。
- その後、同じ Reservoir で異なる位相シフトを持つ系に対しても予測を行う。
 - ▶ 未知の外力に対する Reservoir の予測性能を測る。

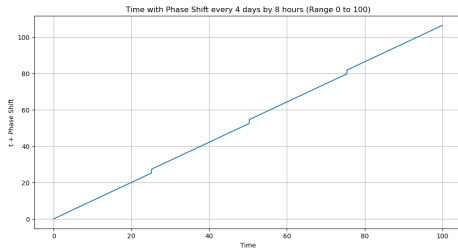


Figure: 4 日に一度 8 時間早めたときの外力の位相

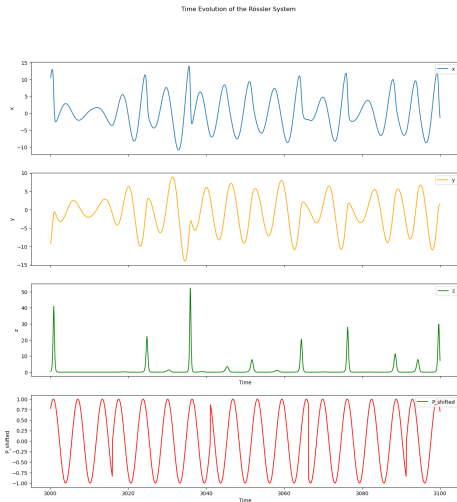


Figure: 位相シフトのある周期外力付きの Rössler システム：上から $x, y, z, P(t)$ 。

Appendix

Reservoir Computer の構造

[1] では, Reservoir Computer が非線形力学系に対して良いモデル・予測性能を持つことに関して, 理論的な説明を試みている.

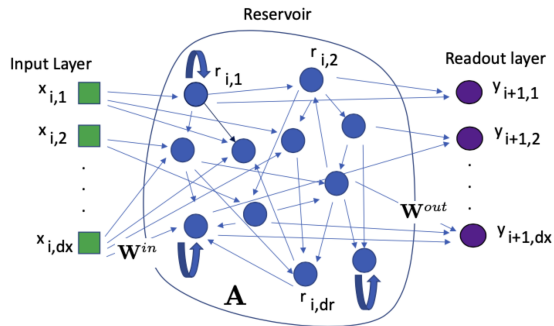


Figure: Reservoir Computer の各層の関係
Fig. 2 from [1]

Reservoir に関する Hyperparameters

Reservoir に関する Hyperparameter のうち、最適化の対象としたものを記した。説明は、[5] の Tutorial 4 を参考にした。

- ① Cell Number (N): セル数.
- ② spectral radius (sr): スペクトル半径. Reservoir の 行列 W の固有値の絶対値の最大値.
 - ▶ 小さいと安定したダイナミクス、大きいとカオス的なダイナミクス.
 - ▶ 理論的には 1 に近いと Reservoir の初期条件に影響を受けにくく、良好な memory を持つことが推測される.
- ③ input scaling (iss): Reservoir の入力 W_{in} に適用される係数. 入力にゲインを加える.
 - ▶ 高くすると Reservoir と 入力の相関を（飽和点まで）高める.
 - ▶ 低くすると Reservoir は外部の入力より自身のダイナミクスに強く影響を受ける.
 - ▶ 多変量時系列データの各変数の影響度を調整可能.
- ④ leaking rate (lr): 次ステップの決定に際しての現在の状態と新しい入力の影響度の割合.
 - ▶ 高い（低い）と惰性が高く（低く）、過去の記憶状態が高く（低く）なる.
 - ▶ ESN のネットワークがその状態を変化させる速度を制御する.
- ⑤ ridge (ridge):

Hyperparameters 最適化のアルゴリズム

Hyperparameters の最適化には python ライブラリである Optuna を用いた。
Optuna で実装されている Hyperparameters の最適化アルゴリズムは例えば以下がある。

`optuna.samplers: from [4]`

- ① **optuna.samplers.RandomSampler:** ランダムサンプリングを使用するサンプラー。
- ② **optuna.samplers.TPESampler:** TPE (Tree-structured Parzen Estimator) アルゴリズムを使用するサンプラー。
- ③ **optuna.samplers.CmaEsSampler:** CMA-ES (Covariance Matrix Adaptation Evolution Strategy) をバックエンドとして使用するサンプラー。

- この研究では, CmaEsSampler (最終的に採用), TPESampler と hyperopt という別の python ライブラリの random search を用いた。
 - ▶ 経験的に, その順により好ましい best value を与えた。
- また, Optuna 搭載の pruner も利用した。
 - ▶ 見込みの悪い Hyperparameters のセットを見限ること。
 - ▶ 使用したのは SuccessiveHalvingPruner。
- なお, hyperopt には Optuna のいくつかの機能は搭載されていない。

- Rössler 系の数値シミュレーション

- ▶ $A = 1.0$
- ▶ $a = 0.2, b = 0.2, c = 5.7$.
- ▶ 初期条件: $[X_0, Y_0, Z_0] = [1.0, 1.0, 1.0]$
- ▶ 時間範囲: $t_{\text{span}} = [0, 4510]$.
- ▶ 位相シフト時間: $p = 8$ に対して Hyperparameter を最適化.
- ▶ 注. 配列データは 1 タイムステップに対して 10 個データポイントをとって生成したので, 長さ $[0, 45100]$ である.

- Optuna の最適化

- ▶ nb seeds = 3
- ▶ nb trials = 3000
- ▶ 最適化アルゴリズム: CmaEsSampler
- ▶ pruner: SuccessiveHalvingPruner
- ▶ train len = 20000
- ▶ test len = 10000

- Hyperparameters の探索空間

- ▶ $N = 10000$: 固定
- ▶ sr: (1e-2, 10, log = True)
- ▶ lr: (1e-3, 1, log = True)
- ▶ iss: (0, 1)
- ▶ ridge: (1e-9, 1e-2, log = True)
- ▶ 一様ランダムにサンプリング.
- ▶ log = True で log をとって一様ランダムにサンプリング.

- 採用した Hyperparameters のセット

- ▶ Best value: 0.0017332259260817507
- ▶ Best parameters:
'sr': 0.568437354122632,
'lr': 0.33989147591891816,
'iss': 0.08827385538440446,
'ridge': 1.30084237042553e-08

参考文献 - 1

- [1] Bollt, E.M. (2020). On explaining the surprising success of reservoir computing forecaster of chaos? The universal machine learning dynamical system with contrast to VAR and DMD. *Chaos*, 31 1, 013108 .
- [2] Trouvain, N., Pedrelli, L., Dinh, T. T., Hinaut, X. (2020) Reservoirpy: an efficient and user-friendly library to design echo state networks. In *International Conference on Artificial Neural Networks* (pp. 494-505). Springer, Cham.
- [3] Yamaguchi, Y., Suzuki, T., Mizoro, Y., Kori, H., Okada, K., Chen, Y., Fustin, J. M., Yamazaki, F., Mizuguchi, N., Zhang, J., Dong, X., Tsujimoto, G., Okuno, Y., Doi, M., and Okamura, H. (2013). Mice genetically deficient in vasopressin V1a and V1b receptors are resistant to jet lag. *Science* (New York, N.Y.), 342(6154), 85 – 90.

参考文献 - 2

- [4] Optuna. (2018). Optuna Documentation. Retrieved from <https://optuna.readthedocs.io/en/stable/> [Accessed on: December 7, 2023]
- [5] ReservoirPy Team. (Year). ReservoirPy Documentation. Retrieved from <https://reservoirpy.readthedocs.io/en/latest/> [Accessed on: December 7, 2023].