# 卒論　ノート

氏名：久野証

所属：東大工学部計数工学科数理情報工学コース

学籍番号: 03-210599

2023 年 10 月 2 日

## 目次



**概要**

2023 年 A セメスターの卒論執筆に際して、勉強したことや考えたことをここにまとめた。

# 1 Papers

## 1.1 Emergence of a resonance in machine learning

Zheng-Meng Zhai , 1 Ling-Wei Kong , 1 and Ying-Cheng Lai 1,2,* 1School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, Arizona 85287, USA 2Department of Physics, Arizona State University, Tempe, Arizona 85287, USA.

(Received 9 June 2022; revised 1 March 2023; accepted 26 July 2023; published 24 August 2023)

キーワード：Resonance in nonlinear dynamical systems,

### 1.1.1 要旨

1. 入力信号にノイズを挿れた場合の Reservoir Computing を考える。
   (a) [47] によれば、training phase と testing phase のノイズ振幅が同じ時 RC は最も良いパフォーマンスを発揮する。
   (b) hyperparameters が最適化されていない時でも、ノイズを挿れることで予測の精度をあげることができる。
   (c) もっとも良い精度を達成するには、hyperparameters が最適化されていなければならない。
      i. Hyperparameters に対する Baysian optimization で最適化可能。
      ii. 確率共鳴があると決定づけるために、ノイズの振幅を hypermparameter に数える。
2. Macky-Glass (MG) system と Kuramoto-Sivashinsky (KS) system に対して、シミュレーションを行う。
3. 機械学習の力学系に対して物理的なモデルを用いて、機械学習における確率共鳴が生まれる原理を考える。

### 1.1.2 手法

1. MG/KS system に対して、Baysian Optimization で最適な hyperparameters（$\sigma$ を含む）を決定する。
2. $\sigma$ を最適値から両側に動かして、$\sigma$ ごとに他の hyperparameters の最適値を決定する。
3. $\sigma$ について、予測誤差を観察することによって、確率共鳴があることを確かめる。

### 1.1.3 状況設定

Appendix A 参照。

1. hyperparameters:
   (a) $\rho$: the specral radius of the reservoir network.
   (b) $\gamma$: the scaling factor of the input weights.
   (c) $\alpha$: the leakage parameter
   (d) $\beta$: the regularization coefficient
   (e) p: the link connection probability of the random network in the hidden layer.
   (f) $\sigma$: the noise amplitude, taking vallue between $[10^{-8}, 10^{0.5}]$.
2. hyperparameters の最適化
   (a) MATLAB: SURROGATEOPT を用いる。[*1]
   (b) $\sigma$ ごとに hyperparameters の最適化を行うので、$\sigma$ に対する他の hyperparameters の組みは異なる。
3. シミュレーションを行う。
   (a) MG system:
   $$\dot{s}(t) = \frac{as(t-\tau)}{(1+[s(t-\tau)]^c) - bs(t)},$$
   $\tau$ is the time delay, $a, b$, and $c$ are parameters. [*2]
      i. $a = 0.2, b = 0.1,$ and $c = 10$ を固定。
      ii. $\tau = 17, 30$ の 2 つの場合を比べる.
      iii. 時系列データには事前に z-score normalization: $z(t) = [s(t) - \bar{s}]/\sigma_s$ を施す。
   (b) KS system:
   $$\frac{\partial u}{\partial t} + \mu\frac{\partial^4 u}{\partial x^4} + \phi\left(\frac{\partial^2 u}{\partial x^2} + u\frac{\partial u}{\partial x}\right) = 0$$
      i. $u(x,t)$ is a scalar field defined in the spatial domain $0 \leqslant x \leqslant L$.
      ii. $\mu = 1$ and $\phi = 1$ , and use the periodic boundary condition.

[*1] "The Bayesian optimization method can be implemented using PYTHON or other languages. Different packages for Bayesian optimization are now available, such as BAYESIAN-OPTIMIZATION and BOTORCH in PYTHON."としている。

[*2] The state of the system at time t is determined by the entire prior state history within the time delay, making the phase space of the system infinitely dimensional.

iii. $L = 60$, where the system has seven positive Lyapunov exponents:

$\lambda_+ \approx 0.089, 0.067, 0.055, 0.041, 0.030, 0.005$, and $0.003$.

4. それぞれに対する ESN の変数設定は表 1 にある。

| | MG | KS |
|---|---|---|
| Warmup | $10000\Delta t$ | 300 Lyapunov times |
| Training phase | $150000\Delta t$ | 1000 Lyapunov times |
| Testing phase for Bayesian optimization | $900\Delta t$ | – |
| Short-term prediction | $900\Delta t$ | 6 Lyapunov times |
| Long-term prediction | $20000\Delta t$ | 200 Lyapunov times |

表 1　論文中の ESN に関する変数設定

#### 1.1.4　面白いと思ったところ

1. ノイズを入れることによって、本来初期値鋭敏性を持つカオスシステムに対して、短期的にも長期的にも予測の精度を挙げられる。
2. Bollt[40] の結果を用いることで、本来 high-dimensional neural network の複雑な dynamics に対して簡略化された物理モデルを得ることができ、それに対する解析によって RC の hidden layer の中身を説明できる。

   $\longrightarrow$ 同じ手法で、この物理モデルを用いることで、他の現象が neural network の dynamics においても現れることを発見・説明できるかもしれない？

#### 1.1.5　疑問点

1. Fig.4 と Fig.5 について，MG system における $\tau$ の値が 30 から 17 に変えると，$\sigma$ にどのような影響があるか．なぜその影響が生まれるか.
2. なぜ，Fig.2 の（逆）ピークを与える $\sigma$ 帯と Fig.6(c) の（逆）ピークを与える $\sigma$ 帯が重なるのか.
3. III で，Machine learning における resonance が生まれる Physical reason を挙げているが，これは対象を正しく説明できているか．extraordinarily complicated な hidden layer の中身を解析することなく，physical reason を与えることが，なにを説明しているのか/なにを説明していないのか.

   $\longrightarrow$ Bollt[40] の結果？

4. そもそも、shor/long-term prediction の期間は何を表しているのか。

   $\longrightarrow$ ある時刻 $t$ の系の状態は時刻 $t-\tau$ から $t$ までの情報で決定される。short/long-term に依らず定まった時間の warmup と training phase を設ける。（ここからは調べる必要あり）これによって、ESN は学習を済ませた状態になる。short/long-term の testing phase の長さの分だけ学習済みの ESN に予測をさせ、実際のデータとの誤差を計測する。

5. $T_{opt}$ は ESN の学習のどの段階に設けられるのか。Training phase の後？

6. $\sigma$: noise amplitude 以外のパラメータの取りうる値の範囲は？

7. Baysian optimization では何について最適化を行なっているのか？

   $\longrightarrow$KS システムの short-term prediciton は RMSE, horizon prediction, stability を観察しているが、全て RMSE を基にしている。よって、最適化は RMSE を目的関数にとっているのではないか（reservoirpy のチュートリアルでは R2 という距離も用いられている）。

8. RMSE, horizon prediction, stability でピークを与える $\sigma$ が同じなのは、これらの手法の intrinsic な性質として導かれることか。それぞれの場合を切り離して計測することでどのような新しいことが言えるか。

#### 1.1.6 論文を受けての今後の研究方向

1. short/long-term prediction での最適なパラメータは同じか、否か。

   （a）実験手法：short/long-term prediction それぞれに対して Baysian optimization でパラメータの最適化を行い、一致するかどうか確かめる。

   （b）本文中では記載がないように思えるが、前提としている先行研究で明らかにされているかもしれない。

2. （些末な点？）論文中では、予測誤差をいくつかの方法で表している（KS に対する short-term prediction だと RMSE, horizon prediction, stability 等）。Baysian optimization の目的関数はどのように決まるか。目的関数の取り方によって、最適な hyperparameters はどのように変化するか。

3. （機械学習の知識が足りていない）論文中では、MG, KS に対して、表 2 のようなパラメータでを定数として hyperparameters の最適化を行なっている。hyperparameters ごとのこれらのパラメータの最適値は（大きく）変化しないのか。

4. ノイズの入れ方: RC の hidden layer の入力信号に対するガウシアンノイズ以外に、どのようなノイズの入れ方と結果が考えられるか。

5. Bollt[40] の手法を理解し、力学系の理論で他に neural network の力学系に応用できる結果がないか探す。

| | MG | KS |
|---|---|---|
| Warmup | $10000\Delta t$ | 300 Lyapunov times |
| Training phase | $150000\Delta t$ | 1000 Lyapunov times |
| Testing phase for Bayesian optimization | $900\Delta t$ | – |
| Short-term prediction | $900\Delta t$ | 6 Lyapunov times |
| Long-term prediction | $20000\Delta t$ | 200 Lyapunov times |

表 2　論文中の ESN に関する変数設定（再掲）

#### 1.1.7　関連する文献

1. Bollt[40]: Dynamics in nerual network の簡易物理モデルを与える。
2. [51]:
3. [57, 58]: Langevin 方程式に関する確率共鳴の理論。

#### 1.1.8　用語まとめ

#### 1.1.9　Abstract

stochastic/coherence resonance, nonlinear dynamical system, regularizer/regularization, reservoir computing, state variables/attractor, hyperparameters.

#### 1.1.10　I. Introduction

model-free/data-driven, oscillation/Lyapunov times, trajectory, basin boundary, robustness, Bayesian optimization.

#### 1.1.11　II. Result

SURROGATEOPT function (MATLAB), surrogate approximation function, objective function, global minimum, sampling/updating, radial basis function, Mackey-Glass (MG) system, spatiotemporal chaotic Kuramoto-Sivashinsky (KS) system.

#### 1.1.12 A. Emergence of a resonance from short-term prediction

transient behavior, z-score normalization, periodic boundary condition, Prediction horizon/stability.

#### 1.1.13 B. Emergence of a resonance from long-term prediction

collapse, wider/narrower resonance.

#### 1.1.14 III. HEURISTIC REASON FOR THE OCCURRENCE OF A RESONANCE

time-scale match, the mean first-passage time, nonlinear activation, linear reservoir computing, noise-enhanced temporal regularity, vector autoregressive process (VAR).

#### 1.1.15 IV. DISCUSSION

magnitude.

#### 1.1.16 Appendix A

recurrent neural network(RNN), input/hidden/output layer, linear regression, adjacency matrix, state vector, dynamical state/evolution, neuron, leakage parameter $\alpha$, link probability p, spectral radius.

## 1.2 [Bollt] On Explaining the Surprising Success of Reservoir Computing Forecaster of Chaos? The Universal Machine Learning Dynamical System with Contrasts to VAR and DMD

RC が重みをランダムに選んでいるのにうまくいく理由は明らかにされていない。ここでは、単純な場合、internal activation function が恒等関数である場合の RC にこの問題を限定し、次の方法でこの問題の説明を試みる。

- 特別な場合の RC に対して WOLD の理論を含む VAR（Vector Autoregressive Averages)、特に NVAR の理論を適用する。
- これらのパラダイムを DMD(Dynamic Mode Decomposition) と紐付ける。

### 1.2.1 1. Introduction

1. 従来の NN 手法の問題点
   (a) Back propagation を用いる Artificial neural networks (ANN): データの最適化に関して計算量が極めて多い
   (b) RNN, LSTM: 短期的なデータに対しては有効だが、完全な学習に関しては高級。
2. RC/ESN: 出力層だけの学習で効率がいい。
3. RC を activation function が線形であるときに限定することで、より成熟した理論の適用を可能にする。
   (a) ARMA:AR (Thoery of autoregression) from time-series analysis and MA (moving averages).
   (b) WOLD:
   (c) VAR (Vector autoregression):
   (d) VMA (Vector moving averages):
   (e) DMD (Dynamic mode decomposition): empirical formulation of Koopman spectral theory.
4. The machine learning RC approaches, econometrics time-series VAR approach, and also the dynamical systems operator theoretic DMD approach の統合。
5. 2 (1.2.2) $\longrightarrow$ 3 (1.2.3) $\longrightarrow$ 4 $\longrightarrow$ 5 $\longrightarrow$ 7 $\longrightarrow$ 8 $\longrightarrow$ 9 $\longrightarrow$ 6.

### 1.2.2 2. The Data as Sampled From a Stochastic Process

### 1.2.3 3. Review of The Traditional RC With Nonlinear Sigmoidal Activation Function

1. Training data: $\{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^{d_x}$
2. The hidden variable: $\mathbf{r}_i \in \mathbb{R}^{d_r}$
   (a) $d_r > d_x$.
3. The reservoir computing RNN:

$$\mathbf{r}_{i+1} = (1-\alpha)\mathbf{r}_i + \alpha q\left(\mathbf{A}\mathbf{r}_i + \mathbf{u}_i + \mathbf{b}\right)$$
$$\mathbf{y}_{i+1} = \mathbf{W}^{\text{out}}\ \mathbf{r}_{i+1}$$

   (a) $\mathbf{A} : d_r \times d_r,\ \mathbf{A}_{i,j} \sim U(-\beta, \beta)$, with $\beta$: the spectral radius.
   (b) $\mathbf{W} : d_r \times d_x,\ \mathbf{W}^{in}_{i,j} \sim U(0, \gamma)$, with $\gamma > 0$: the inner variables $\mathbf{r}$.
   (c) $\mathbf{u}_i = \mathbf{W}^{in}\mathbf{x}_i$.
   (d) $\mathbf{W}^{out} : d_x \times d_r$, readout.
   (e) $q : \mathbb{R} \to \mathbb{R}$: "activation" function.
   (f) $\alpha = 1 (0 \leq \alpha \leq 1)$.
   (g) $\mathbf{b} = 0$.
4. 次の式で $\mathbf{W}_{\text{out}}$ を学習する（線形)。$\mathbf{R} = [\mathbf{r}_{k+1} | \mathbf{r}_{k+2}| \dots | \mathbf{r}_N], k \geq 1$ として、

$$\mathbf{W}_{\text{out}} = \underset{\mathbf{V}\in\mathbb{R}^{d_x \times d_r}}{\arg\min} \|\mathbf{X} - \mathbf{V}\mathbf{R}\|_F = \underset{\mathbf{V}\in\mathbb{R}^{d^x \times d_r}}{\arg\min} \sum_{i=k}^{N} \|\mathbf{x}_i - \mathbf{V}\mathbf{r}_i\|_2,\ k \geq 1^{*3}.$$

   即ち、

$$\mathbf{X} = [\mathbf{x}_{k+1} | \mathbf{x}_{k+2}| \dots | \mathbf{x}_N] = [\mathbf{V}\mathbf{r}_{k+1} | \mathbf{V}\mathbf{r}_{k+2}| \dots | \mathbf{V}\mathbf{r}_N] = \mathbf{V}\mathbf{R},\ k \geq 1$$

   なる $\mathbf{V}$ を求める。
   (a) ridge regression (Tikhonov regularization) により、

$$\mathbf{W}^{\text{out}} := \mathbf{X}\mathbf{R}^T\left(\mathbf{R}\mathbf{R}^T + \lambda\mathbf{I}\right)^{-1}$$

   ただし、$\lambda \geq 0$.
   (b) $\mathbf{R}^\dagger_\lambda := \mathbf{R}^T\left(\mathbf{R}\mathbf{R}^T + \lambda\mathbf{I}\right)^{-1}$ とする（擬似逆行列)。
5. パラメータの取り方に関してはいくつかの問題が残っている（1.2.5)。

*3 k はメモリに関わってくる定数。途中からでも良いということ。この値が k によらないことを示す。

### 1.2.4 4. RC With A Fully Linear Activation, q(s) = s, Yields a VAR(k)

1.

$$\begin{aligned}
\mathbf{r}_{k+1} &= \mathbf{A}\mathbf{r}_k + \mathbf{u}_k \\
&= \mathbf{A}\left(\mathbf{A}\mathbf{r}_{k-1} + \mathbf{u}_{k-1}\right) + \mathbf{u}_k \\
&\vdots \\
&= \mathbf{A}^{k-1}\mathbf{W}^{in}\mathbf{x}_1 + \mathbf{A}^{k-2}\mathbf{W}^{in}\mathbf{x}_2 + \ldots + \mathbf{A}\mathbf{W}^{in}\mathbf{x}_{k-1} + \mathbf{W}^{in}\mathbf{x}_k \\
&= \sum_{j=1}^{k} \mathbf{A}^{j-1}\mathbf{u}_{k-j+1} = \sum_{j=1}^{k} \mathbf{A}^{j-1}\mathbf{W}^{in}\mathbf{x}_{k-j+1},
\end{aligned}$$

$$\begin{aligned}
\mathbf{y}_{\ell+1} &= \mathbf{W}^{\text{out}}\ \mathbf{r}_{\ell+1} \\
&= \mathbf{W}^{\text{out}}\ \sum_{j=1}^{\ell} \mathbf{A}^{j-1}\mathbf{W}^{\text{in}}\ \mathbf{x}_{\ell-j+1} \\
&= \mathbf{W}^{\text{out}}\ \mathbf{A}^{\ell-1}\mathbf{W}^{\text{in}}\ \mathbf{x}_1 + \mathbf{W}^{\text{out}}\ \mathbf{A}^{\ell-2}\mathbf{W}^{\text{in}}\ \mathbf{x}_2 + \ldots + \mathbf{W}^{\text{out}}\ \mathbf{A}\mathbf{W}^{\text{in}}\ \mathbf{x}_{\ell-1} + \mathbf{W}^{\text{out}}\ \mathbf{W}^{\text{in}}\ \mathbf{x}_\ell \\
&= a_\ell\mathbf{x}_1 + a_{\ell-1}\mathbf{x}_2 + \ldots + a_2\mathbf{x}_{\ell-1} + a_1\mathbf{x}_\ell
\end{aligned} \tag{1}$$

with notation,

$$a_j = \mathbf{W}^{\text{out}}\ \mathbf{A}^{j-1}\mathbf{W}^{in}, j = 1, 2, \ldots, \ell.$$

$a_j : d_x \times d_x$ matrices.

式 (1) は VAR($k$) の係数行列の表式:

$$\mathbf{y}_{k+1} = c + a_k\mathbf{x}_1 + a_{k-1}\mathbf{x}_2 + \ldots + a_2\mathbf{x}_{k-1} + a_1\mathbf{x}_k + \boldsymbol{\xi}_{k+1}$$

と合致する[*4]。

2.

$$\begin{bmatrix} \mathbf{y}_{k+1} & \mathbf{y}_{k+2} & \ldots & \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} [a_1] & [a_2] & \ldots & [a_k] \end{bmatrix} \begin{bmatrix} \mathbf{x}_k & \mathbf{x}_{k+1} & \ldots & \mathbf{x}_{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_{k-1} & \mathbf{x}_k & \ldots & \mathbf{x}_{N-2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \ldots & \mathbf{x}_{N-k} \end{bmatrix}$$

を

$$\mathbf{Y} = \mathbf{a}\mathbb{X}$$

と書けば、

(a) $\mathbf{a} : d_x \times (kd_x)$

[*4] $\xi$ はノイズ項。

(b) $\mathbf{Y} = \lceil \mathbf{y}_{k+1} | \mathbf{y}_{k+2} | \dots | \mathbf{y}_N \rceil : d_x \times (N-k)$

(c) $\mathbb{X} : (kdx) \times (N-k)$

3. 最小二乗法を考えて、

$$J(\mathbf{a}) = \|\mathbf{Y} - \mathbf{a}\mathbb{X}\|_F + \lambda \|\mathbf{a}\|_F$$

を最小化する $\mathbf{a}^*$ を求めると、

$$\mathbf{a}^* = \mathbf{Y}\mathbb{X}^T \left(\mathbb{X}\mathbb{X}^T + \lambda I\right)^{-1} := \mathbf{Y}\mathbb{X}_\lambda^\dagger$$

で与えられる。

■4.1

### 1.2.5 残された課題

1. Linear RC with quadratic read-out の議論 (1.2.1).
2. $d_r > d_x$ must be "large enough," but how big is not well understood. Furthermore, the nature of the underlying distribution of matrices $\mathbf{W}^{in}$ and $\mathbf{A}$ is not fully understood ... However, we go on in Sec. 6, with details in Appendix 14, to show that fitting a quadratic read-out, that is extending Eq. (8) to also include terms $\mathbf{r} \circ \mathbf{r}$ (componentwise multiplication, "o" is called the Hadamard product) yields a quadratic NVAR of all monomial quadratic terms, which we observe performs quite well (1.2.3).

# 2 Lectures

1. Reservoir Computing
   (a) Reservoir Computing for SDEs(Josef Teichmann:)
   (b) Reservoir Computing & Dynamical Systems - Second Sumposium on Machine Learning and Dynamical Systems(Josef Teichmann)
   (c) Introduction to Next Generation Reservoir Computing(Daniel Gauthier)
2. Machine Learning in general
   (a) Machine Learning in Finance(Josef Teichmann)

## 2.1 Josef Teichmann: Reservoir Computing for SDEs

Access from here.

1. We consider differential equations of the form

$$dY_t = \sum_i V_i\left(Y_t\right) du_t^i, Y_0 = y \in E$$

   to construction evolutions in state space $E$ (could be a manifold of finite or infinite dimension) depending on local characteristics, initial value $y \in E$ and the control $u$.

2. Theorem (Universality) Let Evol be a smooth evolution operator on a convenient vector space $E$ which satisfies (again the time derivative is taken with respect to the forward variable $t$ ) a controlled ordinary differential equation

$$d\mathrm{Evol}_{s,t}(x) = \sum_{i=1}^{d} V_i\left(\mathrm{Evol}_{s,t}(x)\right) du^i(t).$$

   Then for any smooth (test) function $f : E \to \mathbb{R}$ and for every $M \geq 0$ there is a time-homogenous linear $W = W\left(V_1, \ldots, V_d, f, M, x\right)$ from $\mathbb{A}_d^M$ to the real numbers $\mathbb{R}$ such that

$$f\left(\mathrm{Evol}_{s,t}(x)\right) = W\left(\pi_M\left(\mathrm{Sig}_{s,t}(1)\right)\right) + \mathcal{O}\left((t-s)^{M+1}\right)$$

   for $s \leq t$

3. Signature as universal dynamical system

(a) This explains that any solution can be represented - up to a linear readout - by a universal reservoir, namely signature. Similar constructions can be done in regularity structures, too (branched rough paths, etc).

(b) This is used in many instances of provable machine learning by, e.g., groups in Oxford (Harald Oberhauser, Terry Lyons, etc), and also ...

(c) ... at JP Morgan, in particular great recent work on 'Nonparametric pricing and hedging of exotic derivatives' by Terry Lyons, Sina Nejad and Imanol Perez Arribas.

(d) in contrast to reservoir computing: signature is high dimensional (i.e. infinite dimensional) and a precisely defined, non-random object.

(e) Can we approximate signature by a lower dimensional random object with similar properties?

# 3 TODOs

1. reservoirpy 関連
   (a) ~~Week 2: Understand and optimize ESN hyperparameters などの Tutorial ページを読む.~~
2. Reservoir Computing について学ぶ.
   (a) ~~Week 2: レクチャーノートなどを通じて、知識を準備する。~~とりあえずざっとは見た。
3. ~~Week 2 Github 環境を整備する。~~ 9/28, Week 2: branch の作り方がよくわからないが、とりあえず push は出来た。
4. 論文 1.1 関連
   (a) ~~9/28, Week 2: Bollt[40] を読む。~~一応終わったが、あまり理解できていない。
   (b) Week 3: Bollt[40] をもう一度読む。
   (c) Week 3: Reservoir Computer に関する具体的な問題を見つけるため、情報収集する。
   (d) Week 3: とりあえず何か実装してみる。
      i. 9/28, Week 2: DV の実装。
      ii. rx1 を用いて、並列で動かしてみる。
         A. 並列といってもどこを並列処理させる？ 異なる $\sigma$ に対する他の hyperparameters の最適値を求めるとき？
         B. $\sigma$ を動かしてみる。
   (e) Week 3: ノイズの入れ方を工夫してみる。
   (f) Week 3: ローレンツモデルに対して RC を用いていみる。
5. Week 3: 今後の研究対象を決める。
   (a) 例えば、論文 1.1 から着想を得て、カオスに対する RC を考察するとする。しかし、カオスに対する機械学習の問題意識が予測精度の向上（と計算時間の短縮である）とするならば、その意味で新たに研究すべき対象は見つけにくいのではないか。
   (b) 論文 1.2 にあるとおり、RC はより広いシステムに対する研究も行われているようである。ひとまず、そこまで考える対象をある程度広げてしまった方が出発しやすいのではないだろうか。
   (c) 一貫して機械学習の素養が足りていないため、機械学習そのものの内容というよりは、今は物理・数学的な話題から刺激/触発された RC という文脈で考えておきたい（そのうち RC や機械学習自体の知識が蓄

えられ、少し状況が変わるかもしれない)。