

学習過程に影響を及ぼすネットワークダイナミクスの複雑さに関する一考察

Effect of Complexity on Learning Process of Recurrent Neural Networks

○本間 経康*, 阿部 健一*, 竹田 宏**

○Noriyasu Honma*, Kenichi Abe*, Hiroshi Takeda**

*東北大学, **東北学院大学

*Tohoku University, **Tohoku Gakuin University

キーワード : リカレントニューラルネットワーク (recurrent neural networks), 複雑系 (complex adaptive systems), カオス (chaos), カオスの辺縁 (edge of chaos), 学習 (learning)

連絡先 : 〒980-77 仙台市青葉区星陵町 2-1 東北大学 医療技術短期大学部 診療放射線技術学科
本間 経康, Tel.: (022)717-7940, Fax.: (022)717-7944, E-mail: honma@abe.ecei.tohoku.ac.jp

1. はじめに

人工生命を含めた複雑系(複雑適応系)の研究がブームともいえるほど活発になり, また注目されたのは, 複雑な現象すべてに共通する統一理論の構築が可能であるという希望的予測によるところが大きい⁶⁾. この統一理論はまだ確立されていないが, もし確立されるとすればその核となりうる幾つかの重要な概念は提案されており, その一つが「カオスの辺縁」(または「カオスの縁」, the edge of chaos)である¹⁰⁾.

「カオスの辺縁」は, 複雑系の定義の一つであり, そこでは「計算容量が最大化する」ことを筆頭に, 「生物進化の原動力の一つである」⁷⁾ というスケールの大きな主張まで, さまざまな特徴的性質が報告されている. しかし, 現象論的な側面が注目される一方で, そのメカニズムに関する理論

的背景が弱く, 得られている成果に比べて主張が過大であるとの批判もある⁶⁾.

著者らは, 離散値系および連続値系のリカレントネットワークにおいて, ネットワークダイナミクスを制御するための新たなパラメータを提案し^{3, 8)}, これを用いて「カオスの辺縁」の特徴的性質を利用したシステム同定問題および未知システムの時系列推定問題への応用を試みてきた. とくに, システム制御分野への応用を目指して, 複雑系の研究で主流となっている離散値系モデルを拡張して, リカレントニューラルネットワーク(以下, RNNと記す)を用い, ダイナミクスの複雑さと要素ニューロン間の相互作用の強さなどの関係を調べることにより, 連続値系における「カオスの辺縁」の性質を明らかにした^{4, 9)}. しかし, 統計的な関係を用いたパラメータ改変量の定量化に

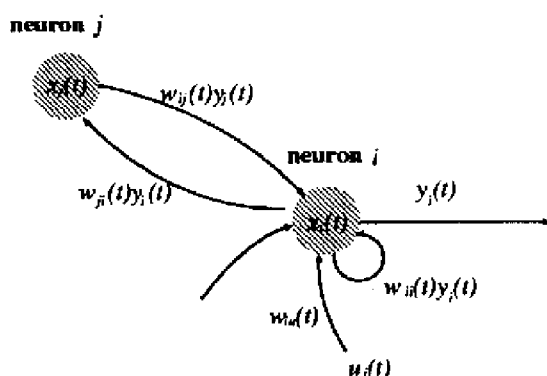


Fig. 1 Fully connected recurrent neural networks

関して成功したもの⁵⁾、これらの試みも現象面での有効性はシミュレーション等で確認できたが、何故有効なのかという理論的考察が十分ではなく、他の複雑系の研究と同じ問題を抱えている。

本稿では、「何故カオスの辺縁が有効なのか？」という基本的な疑問に対する新たな知見が得られたので報告する。また、これに基づいて改良したRNNの新たな学習法を提案する。はじめに、学習対象とRNNのダイナミクスの複雑さの関係を基に、「カオスの辺縁」における学習能力の最大化の理由について理論的に考察し、これに基づくより効率的な学習法を提示する。また、提案学習法を実現するためのRNNのリアプノフ指数の計算式を導出し、具体的な学習法を示す。

2. リカレントニューラルネットワークのダイナミクス

2.1 RNNモデル

システムの振舞いの複雑さは、システムを構成する各要素の振舞いの複雑さのみではなく、要素間の相互作用にも依存する²⁾との立場から、本研究では様々な相互作用（多様な複雑さのダイナミクス）を表現できるモデルとして、各ニューロンの写像関数がシグモイド関数で、次式で時間発展してゆく Fig. 1に示すような結合トポロジー制限のないリカレントニューラルネットワークを考

える。

$$x_i(t+1) = \frac{1}{1 + \exp(-2s_i(t+1)/s_0)}, \quad (1)$$

$$s_i(t+1) = \sum_j^N w_{ij} x_j(t) + \theta_i, \quad (2)$$

$$\hat{y}_i(t) = x_i(t), \quad (t = 1, 2, \dots). \quad (3)$$

ここで、 x_i 、 s_i 、 \hat{y}_i はニューロン*i*の状態、入力、出力で、 N 、 w_{ij} 、 s_0 、 θ_i は、それぞれ要素数、要素*j*との結合重み係数、写像関数の傾き、および閾値である。

2.2 カオスの辺縁と学習能力

ここでは、 n 階常微分方程式

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)), \quad (4)$$

$$\mathbf{y}(t) = [y_1(t), y_2(t), \dots, y_n(t)]^T,$$

$$\mathbf{f}(\mathbf{y}) = [f_1(\mathbf{y}), f_2(\mathbf{y}), \dots, f_n(\mathbf{y})]^T,$$

のアトラクタをサンプリング周期 Δt でサンプリングした時系列 $\mathbf{y}(t)$ 、 $\mathbf{y}(t+\Delta t)$ 、 $\mathbf{y}(t+2\Delta t)$ 、 \dots をRNNに学習させることを考える。ただし、サンプリング周期 Δt は、アトラクタの周期（ストレンジアトラクタの場合は概周期）よりも十分短いとし、簡単のため(1)~(3)のRNNの時間発展間隔と等しい($\Delta t = 1$)とする。

(4)式のアトラクタの複雑さは、(i) 定点アトラクタ、(ii) 周期アトラクタ、(iii) トーラス、(iv) ストレンジアトラクタ（カオス）の4通りに大別できる。この分類は、リアプノフスペクトル¹⁾を用いて定量化できる。たとえば $n=3$ の場合¹⁾、(i) 定点アトラクタは、リアプノフスペクトルの符号がすべて負 $(-, -, -)$ になり、(ii) 周期アトラクタの場合は $(0, -, -)$ となる。(iii) トーラス（この場合

¹⁾ n 次元のアトラクタには、 n 個のリアプノフ指数(3.2.1を参照)が存在し、大きい順に並べたリアプノフ指数の組 $(\lambda_1, \lambda_2, \dots, \lambda_n)$ をリアプノフスペクトル¹⁾という。なお、上記の分類のうち、(iv)のカオスとそれを区別するだけなら、最大リアプノフ指数 λ_1 のみで可能である。すなわち、 $\lambda_1 > 0$ がカオスで、そうでなければカオスではない。また、未知システムからの時系列であってもリアプノフスペクトルが計算可能である¹⁵⁾。

2-トラス)では, $(0, 0, -)$ となり, (iv)カオスでは, $(+, 0, -)$ となる. ここで, 定点アトラクタ以外はいずれも軌道の進行方向のリアプノフ指数が0であることを注意しておきたい.

はじめに $n = N$ の場合について考えると, RNN に定点を除くアトラクタを学習させるためには, RNN の時間発展方向 ($\mathbf{x}(t+1) - \mathbf{x}(t)$ 方向) の軌道の平均拡大率 (以下, この方向のリアプノフ指数を λ_i と書く) を 0 にする, すなわち「カオスの辺縁」にすると効果的であることが分かる[†]. もちろん, 対象となるアトラクタと RNN のリアプノフスペクトルが一致することも必要であるが, 定点を除くすべての対象に対して $\lambda_i = 0$ は必要条件である. なお, リアプノフ指数は, ある方向の拡大率の時間平均を意味するから, リアプノフスペクトルの一致はアトラクタを学習するための必要条件であって, 必要十分条件ではない. これは, スペクトルだけではなく, 各リアプノフ指数の拡大方向が一致している場合でも同様である.

つぎに, $n < N$ の場合^{*}を考察するため, リアプノフ次元を考える. リアプノフ次元は, リアプノフ指数を用いて以下のように定義され, アトラクタを表現するために必要な次元数 (非整数) を表す. $\sum_{k=1}^i \lambda_k \geq 0$ となる最大の i を j とすると, リアプノフ次元 D_L は,

$$D_L = j + \sum_{k=1}^j \frac{\lambda_k}{|\lambda_{j+1}|} \quad (5)$$

となる. すなわち, j 次元の体積素までは拡大するが, $j+1$ 次元の体積素は縮むからこれ以降の $k = j+2, j+3, \dots, n$ に対応する方向は完全に潰れてしまうと考える. このアトラクタを取めるには j 次元と若干の広がり[‡]で十分であると考えること

[†] 定点だけは $\lambda_i = 0$ では RNN のダイナミクスが複雑過ぎてうまく学習できないことが予想されるが, 定点を学習する問題に RNN を用いる場合は少ないと考えられる.

^{*} 階層型ニューラルネットワーク (以下, FNN と略す) の非線形写像能力は, 中間層ニューロンの数を十分多くすることにより保証されていることを考慮すると, $n \geq N$ では十分な写像能力が得られない場合が多いので, $n < N$ の場合を考えるのが一般的である.

を意味する¹⁾. したがって, $n < N$ の場合, 対象アトラクタと RNN のリアプノフスペクトルはその要素数が異なるが, アトラクタのリアプノフ次元を D_L とし, D を $D-1 < D_L \leq D$ を満たす整数として, RNN の第 D 番までのリアプノフスペクトルが対象アトラクタのスペクトルと一致している場合は, 効果的に学習できることが分かる. 同様に, $n > N$ であっても $D_L \leq N$ であれば学習できる可能性がある. いずれの場合も, $\lambda_i = 0$ は必要条件となり, 「カオスの辺縁」が学習に有効であることが分かる.

なお, 学習対象が未知システムの場合は観測できるのが n 次元アトラクタではなく, その一部 (例えば, $y_1(t), y_1(t+1), \dots$) に限定される場合が多いが, 埋め込み理論¹¹⁾により, n 次元アトラクタを再現できることを考えると, この場合も学習に必要なリアプノフスペクトルの条件は同様である. また, 学習対象が $n-1$ 階の差分方程式の場合も, 学習時系列が n 階常微分方程式のポアンカレ写像¹⁾となる場合は, 同様な議論ができる.

3. 提案学習法

ここでは, 前節での議論を踏まえて著者らがこれまでに提案した学習法の問題点を指摘し, それを改善した学習法を提案する.

3.1 学習法 の概念

1 節で述べたように, 従来の「カオスの辺縁」の性質を利用した学習法は, 「カオスの辺縁にネットワークダイナミクスを保つと, 他の複雑さのダイナミクスをもつ RNN よりも学習に成功する可能性が高い⁸⁾」という, 現象論的な有効性をその根拠としていた^{4, 5, 9)}.

また, 「カオスの辺縁」では Fig. 2 に示すように, 1 時刻前の状態との相互情報量 MI が高くなると同時に, MI が高いものから低いものまでさま

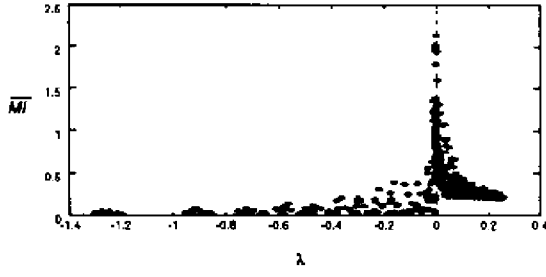


Fig. 2 Average mutual information \overline{MI} versus Lyapunov exponent λ .

さまざまなダイナミクスを表現可能であること、さらに「カオスの辺縁」に保つことによりパラメータの膨大な探索空間を効率的に絞り込めるため、効率良く学習できるという意味づけをしていた。これらの考察はカオスの辺縁の「有効性」の一つ側面を明らかにしているが、前節で得られた「カオスの辺縁は必要条件である」という結果は、より理論的な根拠を与える新しい知見である。

リアプノフスペクトルを完全に一致させるには、 D_L が大きくなると計算量も膨大になることを考慮して、本稿で提案する手法では、最大リアプノフ指数 λ_1 と進行方向のリアプノフ指数 λ_t のみを一致させることを目標とする。すなわち、学習対象の最大リアプノフ指数を λ_1^{obj} とおくと、RNNのリアプノフ指数を $\lambda_1 = \lambda_1^{obj}$ 、 $\lambda_t = 0$ （カオスの辺縁）とするように、ネットワークパラメータ w_{ij}, θ_i を改変する。

ところで、従来手法においては、最大リアプノフ指数を $\lambda_1 \sim 0$ としていたが、前節の必要条件は $\lambda_t = 0$ である。したがって、従来手法で効率的な学習をするためには、 $\lambda_1 \sim \lambda_t$ となっていなければならない。一般には、もちろん $\lambda_1 \neq \lambda_t$ であるが、 N が大きいRNNの場合は、Fig. 3に示すようにほぼ $\lambda_1 \sim \lambda_t$ とみなせることが分かる。

3.2 提案学習法

ここでは、 $\lambda_1 = \lambda_1^{obj}$ 、 $\lambda_t = 0$ とするためのパラメータの改変量を求め、具体的な学習法を提案す

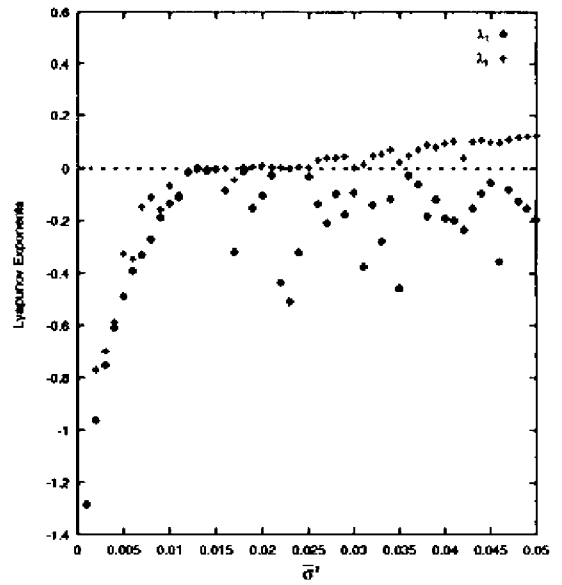


Fig. 3 λ_1 and λ_t

る。はじめに、次の2つの誤差を定義する。

$$e_1 = \frac{1}{2}(\lambda_1 - \lambda_1^{obj})^2, \quad (6)$$

$$e_t = \frac{1}{2}\lambda_t^2. \quad (7)$$

また、学習時系列の個数を T とおき、通常の学習誤差を $e_c = \sum_{t=1}^T (y(t) - \hat{y}(t))^2 / 2$ として、総誤差を $e = e(e_c, e_1, e_2)$ とする。以下では、簡単のため $e = e_c + e_1 + e_2$ とし、 e を減少させるための w_{ij} の改変量 Δw_{ij}^\dagger は、 $\eta > 0$ として次式で求める。

$$\begin{aligned} \Delta w_{ij} &= -\eta \frac{\partial e}{\partial w_{ij}}, \\ &= -\eta \left(\frac{\partial e_c}{\partial w_{ij}} \right. \\ &\quad \left. + (\lambda_1 - \lambda_1^{obj}) \frac{\partial \lambda_1}{\partial w_{ij}} + \lambda_t \frac{\partial \lambda_t}{\partial w_{ij}} \right). \quad (8) \end{aligned}$$

$\frac{\partial e_c}{\partial w_{ij}}$ は、従来のRNNの学習法^{13, 14, 12)}によって計算できるから、 λ_1, λ_t および $\frac{\partial \lambda_1}{\partial w_{ij}}, \frac{\partial \lambda_t}{\partial w_{ij}}$ が分かれば Δw_{ij} を計算できる。以下順番にその計算法を示す。

3.2.1 リアプノフ指数 λ の計算

n 階差分方程式 $\mathbf{x}(t+1) = \mathbf{F}(\mathbf{x}(t))$ の第 i リアプノフ指数 λ_i は、次式で計算される¹⁵⁾。

$$\lambda_i = \lim_{T \rightarrow \infty} \frac{1}{T} \log \left[\frac{\| \mathbf{L}(T) \|}{\| \mathbf{L}(0) \|} \right], \quad (9)$$

[†] θ_i の改変量は省略するが、同様に求められる。

ただし、 ${}^1L(0)$ は、 $\mathbf{x}(0)$ を中心とした無限小の n 次元球体の直径ベクトル、 ${}^1L(T)$ は、その球を T 回遷移させた時に得られる $\mathbf{x}(T)$ を中心とした楕円体の i 番目に大きな主軸ベクトルである。 $\mathbf{x}(t)$ におけるヤコビ行列を $DF(\mathbf{x}(t))$ とおくと、アトラクタ近傍で線形化して、 ${}^1L(t+1) = DF(\mathbf{x}(t)) {}^1L(t)$ となるから、

$$\lambda_i = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \log \left[\frac{\|DF(\mathbf{x}(t)) {}^1L(t)\|}{\|{}^1L(t)\|} \right]. \quad (10)$$

ここで、計算が発散しないように、 ${}^1L(t+1)$ を正規化して

$$\delta {}^1L(t+1) = \frac{DF(\mathbf{x}(t)) \delta {}^1L(t)}{\|DF(\mathbf{x}(t)) \delta {}^1L(t)\|}, \quad (11)$$

とおくと、最大リアプノフ指数を求める場合には、 $\delta {}^1L(0)$ は任意方向の n 次元ベクトルで良い¹⁵⁾ から、 ${}^1L(t) = \|\delta {}^1L(t)\|$ と書くと、次式が得られる。

$$\lambda_1 = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \log {}^1L(t). \quad (12)$$

(1)~(3) の RNN の場合、ヤコビ行列の (i, j) 要素 $DF_{ij}(\mathbf{x}(t))$ は、次式で表される。

$$DF_{ij}(\mathbf{x}(t)) = \frac{2}{s_0} w_{ij} x_i(t+1)(1-x_i(t+1)). \quad (13)$$

したがって、 ${}^1L(t) = [{}^1L_1(t), {}^1L_2(t), \dots, {}^1L_N(t)]^T$ の要素は、次式で表される。

$${}^1L_i(t) = \frac{2}{s_0} X_i(t+1) \sum_{j=1}^N w_{ij} \delta {}^1L_j(t), \quad (14)$$

ただし、 $\delta {}^1L(t) = [\delta {}^1L_1(t), \delta {}^1L_2(t), \dots, \delta {}^1L_N(t)]^T$ とし、 $X_i(t) = x_i(t)(1-x_i(t))$ である。よって、

$${}^1L(t) = \sqrt{\sum_{i=1}^N ({}^1L_i(t))^2}. \quad (15)$$

であるから、 $\|\delta {}^1L(0)\| = 1$ となる任意の n 次元ベクトルを準備すれば、最大リアプノフ指数は、RNN の学習区間の状態時系列 $\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T)$ と結合係数 w_{ij} により計算できる。

つぎに λ_i は、時間的進行方向の拡大率であるから、2.2 で述べたサンプリング間隔が十分に小さい

場合には、以下の近似が成り立つ。

$${}^1L(t) \simeq \mathbf{x}(t+1) - \mathbf{x}(t). \quad (16)$$

したがって、その大きさ ${}^1L(t)$ は、

$${}^1L(t) \simeq \sqrt{\sum_{i=1}^N (x_i(t+1) - x_i(t))^2} \quad (17)$$

であるから、 λ_i は (9) 式より、次式で計算される。

$$\lambda_i = \lim_{T \rightarrow \infty} \frac{1}{T} \log \frac{{}^1L(T)}{{}^1L(0)}. \quad (18)$$

3.2.2 $\partial \lambda / \partial w_{ij}$ の計算

以下では、式の複雑さを避けるため ${}^1L(t)$ を単に $L(t)$ と書く。(12) より、

$$\begin{aligned} \frac{\partial \lambda_1}{\partial w_{ij}} &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{\partial \log L(t)}{\partial w_{ij}}, \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \frac{1}{L(t)} \frac{\partial L(t)}{\partial w_{ij}}. \end{aligned} \quad (19)$$

(15) 式より、

$$\frac{\partial L(t)}{\partial w_{ij}} = \frac{1}{L(t)} \sum_{k=1}^N L_k(t) \frac{\partial L_k(t)}{\partial w_{ij}}. \quad (20)$$

(14) 式より、 Δ_{ik} をクロネッカーのデルタとして

$$\begin{aligned} \frac{\partial L_k(t)}{\partial w_{ij}} &= \frac{2}{s_0} \left(\frac{\partial X_k(t+1)}{\partial w_{ij}} \sum_{m=1}^N w_{km} \delta L_m(t) \right. \\ &\quad \left. + X_k(t+1) [\Delta_{ik} \delta L_j(t) + \sum_{m=1}^N w_{km} \frac{\partial \delta L_m(t)}{\partial w_{ij}}] \right). \end{aligned} \quad (21)$$

ここで、

$$\frac{\partial X_k(t)}{\partial w_{ij}} = (1 - 2x_k(t)) \frac{\partial x_k(t)}{\partial w_{ij}} \quad (22)$$

であり、 $\partial x_k(t) / \partial w_{ij}$ は、従来の RNN の学習アルゴリズム^{13, 14, 12)} により計算できる。また、

$$\begin{aligned} \frac{\partial \delta L_m(t)}{\partial w_{ij}} &= \frac{1}{L^2(t-1)} \left(\frac{\partial L_m(t-1)}{\partial w_{ij}} L(t-1) \right. \\ &\quad \left. - \frac{\partial L(t-1)}{\partial w_{ij}} L_m(t-1) \right) \end{aligned} \quad (23)$$

である。 $\delta L_m(0)$ は、 w_{ij} に関係なく決定できるから、 $\partial \delta L_m(0) / \partial w_{ij} = 0$ とおくと、(21) より $\partial \delta L_k(1) / \partial w_{ij}$

が求まり, (20)式より $\partial L(1)/\partial w_{ij}$ が計算できる. これより, (23)式を用いて $\partial \delta L_m(1)/\partial w_{ij}$ が計算できるから, 以下同様に $\partial L(t)/\partial w_{ij}$ が計算でき, $\partial \lambda_1/\partial w_{ij}$ が求まる.

つぎに, $\partial \lambda_t/\partial w_{ij}$ を求める. (18)式より,

$$\frac{\partial \lambda_t}{\partial w_{ij}} = \lim_{T \rightarrow \infty} \frac{1}{T} \left(\frac{1}{{}^t L(T)} \frac{\partial {}^t L(T)}{\partial w_{ij}} - \frac{1}{{}^t L(0)} \frac{\partial {}^t L(0)}{\partial w_{ij}} \right). \quad (24)$$

(17)式より,

$$\frac{\partial {}^t L(t)}{\partial w_{ij}} = \frac{1}{{}^t L(t)} \sum_{k=1}^N {}^t L_k(t) \left(\frac{\partial x_k(t+1)}{\partial w_{ij}} - \frac{\partial x_k(t)}{\partial w_{ij}} \right). \quad (25)$$

であるから, 同様に $\partial x_k(t)/\partial w_{ij}$ を従来の学習法を用いて求めることにより, $\partial \lambda_t/\partial w_{ij}$ を計算できる.

4. おわりに

本稿では,「カオスの辺縁が何故 RNN の学習に有効なのか」という基本的な疑問に対して, 常微分方程式のアトラクタの学習においては,「カオスの辺縁は必要条件である」という理論的な根拠を与えた. また, この理論を背景として, 従来の「カオスの辺縁」の性質を利用する学習法を改善したアルゴリズムを提案した. 提案アルゴリズムの計算量的評価, およびその収束速度などの評価は今後の課題である.

参考文献

- 1) 合原 一幸: カオス, サイエンス社 (1990)
- 2) 本間, 佐藤, 阿部, 竹田: 自律分散的適応手法によるホロンネットワークの進化について, 計測自動制御学会論文集, 31-7, 908/915 (1995)
- 3) 本間, 佐藤, 阿部, 竹田: ホロンネットワークの創発的進化による非線形システムのダイナミクス推定, 計測自動制御学会論文集, 31-10, 1739/1745 (1995)
- 4) N. Honma, K. Kitagawa and K. Abe: Effect of Complexity on Learning Ability of Recurrent Neural Networks, Proc. of the 2nd AROB, 58/61 (1997)

- 5) N. Honma, K. Kitagawa, K. Abe and H. Takeda: An Autonomous Criterion of Learning Methods for Recurrent Neural Networks, Proc. of the 2nd ASCC, II, 219/222 (1997)
- 6) J. Horgan: From Complexity to Perplexity, Scientific American, June, 74/79 (1995)
- 7) S. A. Kauffman: Antichaos and Adaptation, Scientific American, August, 64/70 (1991)
- 8) 喜多川, 本間, 阿部: カオスの辺縁におけるリカレントニューラルネットワークの学習能力, システム情報関連合同シンポジウム前刷, 103/106 (1995)
- 9) 喜多川, 本間, 阿部: リカレントニューラルネットワークの創発的学習手法, 計測自動制御学会論文集, 33-11 (1997 掲載予定)
- 10) C. G. Langton: Computation at the Edge of Chaos: Phase Transitions and Emergent Computation, Physica D 42 12/37 (1990)
- 11) N. H. Packard, J. P. Crutchfield, et.al: Geometry from a Time Series, Phys. Rev. Lett. 45, 712 (1980)
- 12) B. A. Pearlmutter: Learning State Space Trajectories in Recurrent Neural Networks, Neural Computation, 1-2, 263/269 (1989)
- 13) F. J. Pineda: Generalization of Back-Propagation to Recurrent Neural Networks, Physical Review Letters, 59-19, 2229/2232 (1987)
- 14) R. J. Williams and D. Zipser: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, Neural Computation 1-2, 270/280 (1989)
- 15) A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano: Determining Lyapunov Exponents from a time series, Physica, 16D, 285/315 (1989)