神経回路網ダイナミクスの複雑さの制御法[†]

本 間 経 康*・酒 井 正 夫**

阿 部 健 一**・竹 田 宏***

Complexity Control Methods of Dynamics in Recurrent Neural Networks

Noriyasu HONMA*, Masao SAKAI**,
Kenichi ABE** and Hiroshi TAKEDA***

The paper demonstrates that a complexity of dynamics in recurrent networks with N neurons can be controlled by our gradient methods. The complexity, i.e. the Lyapunov exponent, is calculated by observing the state transition for a long-term period T . One of the control methods is based on the conventional learning algorithms for the recurrent networks. This is the method with high-precision, but it requires $O(N^5T)$ expected time. To reduce the expensive run time, we propose another method based on the approximate relation between the complexity and a new parameter of the network configuration reported in our previous papers. This approximation requires only $O(N^2)$ run time. Simulation results show that the first method can control the exponent and that the approximation one can control the exponent under a restriction. The networks can learn not only the target time series, but also the exponent of the target by a combination method which is incorporated the proposed control method into the conventional learning algorithm.

Key Words: recurrent neural networks, complexity, chaos, learning.

1. はじめに

フィードバック結合をもつリカレントニューラルネットワーク (Recurrent Neural Networks: 以下 RNN と略す) は、構成ニューロン数が等しい階層型ニューラルネットワーク (Feedforward Neural Networks: 以下 FNN と略す) に比べ、シナプス結合の総数が多いため自由度が高く、構成ニューロン間の複雑な相互作用が生み出す複雑なダイナミクスを利用した動的システムの同定問題、制御問題などへの応用が期待されている^{10), 11)}。また、Hopfield らによって FNN とは異なる認識機構 (最適化機構) をもつ完全結合型 RNN が提案されており⁵⁾、ノイズ画像の優れた復元能力などが報告されている⁹⁾。Hopfield らの RNN では、シナプス

結合の値が制限されているため、そのダイナミクスは一点へ向かう (ポイントアトラクタ) という意味で非常に単純である。しかし、Freeman らの動物実験の結果によると、生物の認識機構には、カオスダイナミクスおよびカオスの辺縁 (the edge of chaos) のダイナミクスが深く関わっていることが報告されている¹⁵⁾。したがって、生物のようなより柔軟な認識機構を実現するためには、ネットワークダイナミクスの複雑さを考慮することも重要な一つの方法になりうる。

著者らは先に、完全結合型 RNN のダイナミクスの複雑さとネットワークパラメータの定性的な関係について考察し、ある制約条件を設けることで、所望の複雑さをもつネットワークを設計する手法を提案した⁸⁾。また、この手法により設計した適当な複雑さをもつ RNN と FNN の組み合わせにより、従来のネットワークの認識機構に比べ、より生物の認識機構に近いネットワークを構成できる可能性を示した⁷⁾。

一方、RNN に対する学習法は、FNN の学習法を基にしたものが多く、たとえば Jordan 型 RNN⁶⁾ のように教師信号をそのままフィードバック信号にも用いることで、FNN に対する誤差逆伝搬学習法¹⁴⁾を適用できる構造の RNN も提案されている。この場合、学習対象システムの入出力関係を決定する写像関数を近似的に実現することは可能であるが^{2), 3)}、教師信号のフィードバックが必要なため、本稿で後述するように、RNN の出力信号をフィードバックした場合の最大リアプノフ指数¹⁾ (ダイナミクスの複雑さ) が、対象システム

[†] 計測自動制御学会東北支部 第 170 回研究集会で発表 (1997・10)

* 東北大学医療技術短期大学部 仙台市青葉区星陵町 2-1

** 東北大学大学院工学研究科 仙台市青葉区荒巻字青葉

*** 東北学院大学工学部 多賀城市中央 1-13-1

* College of Medical Sciences, Tohoku University., Aoba-ku, Sendai

** Graduate School of Engineering, Tohoku University., Aoba-ku, Sendai

*** Faculty of Engineering., Tohoku Gakuin University., Tagajyo

(Received May 19, 1998)

(Revised November 4, 1998)

の最大リアプノフ指数に近いという保証はない。したがって、未知システムから観測したデータを基に、従来法を用いてそのシステムのモデルを作成した場合などは、モデルシステムを用いて得られた振舞いに関する知見が、対象システムのそれと大きく異なる場合もありうる。

誤差逆伝搬学習法では、誤差を極小にする方向にネットワークパラメータを変化させるが、このとき同時に最大リアプノフ指数も変化する。したがって、最大リアプノフ指数などで表される、対象システムのダイナミクスの複雑さも含めて RNN で近似的に実現するためには、逆伝搬学習によって逐次的に変化するパラメータ値を、所望の複雑さのダイナミクスを生み出すパラメータ値になるように制御する必要がある。著者らは、先の定性的な関係を基に最大リアプノフ指数を逐次的に所望の値に近付ける基本的な手法を示したが、制約条件がネットワークの近似実現能力を妨げる可能性があるなど、その精度は十分であるとはいえない手法であった⁸⁾。

本稿では、一般的なニューロン間の相互作用を実現できる完全結合型 RNN と、写像関数の近似能力に優れる Jordan 型 RNN について、最大リアプノフ指数を逐次的に制御する方法を提案する。また、構成ニューロン数の多い完全結合型 RNN に対する提案手法の高い計算コストを低減するための近似計算法を示し、その性能についてもあわせて報告する。さらに、Jordan 型 RNN の学習過程における最大リアプノフ指数の変化について考察し、提案手法と従来の学習法の組合せにより、最大リアプノフ指数に関しても、対象ダイナミクスの複雑さに、より近いモデルが構築できることを示す。

2. RNN モデルとダイナミクスの複雑さ

本稿では、結合構造に制限がない、すなわちすべてのユニット間が相互に結合されていて、一般的な相互作用を実現できる完全結合型と、結合に制限を加えることで関数近似能力³⁾に優れた Jordan 型について取り上げる。

また、ダイナミクスの複雑さを定量化する指標は、種々提案されているが、本稿ではリアプノフスペクトルを用いる。完全なリアプノフスペクトルを用いればより精密なダイナミクスの複雑さを規定できるが、たとえばカオスであるかどうかは、最大リアプノフ指数のみで判定できる。すなわち、最大リアプノフ指数 λ は、システムの初期値の微小な違いの広がり程度を表す指標で、カオスであれば $\lambda > 0$ となることが知られている¹⁾。したがって、計算コストなども考慮して、以下では最大リアプノフ指数のみを用いることにする。

n 階差分方程式

$$\mathbf{x}(t+1) = \mathbf{F}(\mathbf{x}(t)), \quad (1)$$

$$\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T,$$

$$\mathbf{F}(\mathbf{x}(t)) = [f_1(\mathbf{x}(t)), f_2(\mathbf{x}(t)), \dots, f_n(\mathbf{x}(t))]^T,$$

の最大リアプノフ指数 λ は、次式で計算される¹⁷⁾。

$$\lambda = \lim_{T \rightarrow \infty} \frac{1}{T} \log \left[\frac{\|\mathbf{L}(T)\|}{\|\mathbf{L}(0)\|} \right]. \quad (2)$$

ただし、 $\mathbf{L}(0)$ 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、 $\mathbf{L}(T)$ は、その球を (1) 式で T 回遷移させたときに得られる $\mathbf{x}(T)$ を中心とした楕円体の最大の主軸ベクトルである。 $\mathbf{x}(t)$ におけるヤコビ行列を $\mathbf{DF}(\mathbf{x}(t))$ とおくと、アトラクタ近傍で線形化して、 $\mathbf{L}(t+1) = \mathbf{DF}(\mathbf{x}(t))\mathbf{L}(t)$ となるから、

$$\lambda = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=0}^{T-1} \log \left[\frac{\|\mathbf{DF}(\mathbf{x}(t))\mathbf{L}(t)\|}{\|\mathbf{L}(t)\|} \right]. \quad (3)$$

ここで、分子の計算が発散しないように、 $\mathbf{L}(t)$ を正規化して

$$\mathbf{L}(t+1) = \mathbf{DF}(\mathbf{x}(t))\delta\mathbf{L}(t), \quad (4)$$

とする。ただし、 $\mathbf{L}(t) = \|\mathbf{L}(t)\|$ として

$$\delta\mathbf{L}(t) = \frac{\mathbf{L}(t)}{L(t)}, \quad (5)$$

である。このとき、 $\delta\mathbf{L}(0)$ として大きさが 1 の任意方向の n 次元ベクトルを準備すると、次式が得られる。

$$\lambda = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \log L(t). \quad (6)$$

なお、最大リアプノフ指数だけでなく、完全なリアプノフスペクトルを求める方法も同様に定式化されている¹⁷⁾。

2.1 完全結合型 RNN と最大リアプノフ指数

各要素ニューロンの入出力特性がシグモイド関数で、 i 番目 ($i = 1, 2, \dots, N$) のニューロンが次式で時間発展してゆく完全結合型 RNN を考える。

$$x_i(t+1) = \frac{1}{1 + \exp(-2s_i(t+1)/s_0)}, \quad (7)$$

$$s_i(t+1) = \sum_{j=1}^N w_{ij}x_j(t) + \theta_i, \quad (t = 1, 2, \dots). \quad (8)$$

ここで、 x_i , s_i はニューロン i の出力、入力で、 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、シグモイド関数の傾き、および閾値である。

(7), (8) 式の完全結合型 RNN の場合、ヤコビ行列の (i, j) 要素 $\mathbf{DF}_{ij}(\mathbf{x}(t))$ は、次式で表される。

$$\mathbf{DF}_{ij}(\mathbf{x}(t)) = \frac{2}{s_0} w_{ij} x_i(t+1)(1 - x_i(t+1)). \quad (9)$$

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

$$L_i(t+1) = \frac{2}{s_0} X_i(t+1) \sum_{j=1}^N w_{ij} \delta L_j(t). \quad (10)$$

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

$$L(t) = \sqrt{\sum_{i=1}^N (L_i(t))^2}, \quad (11)$$

であるから、 $\|\delta\mathbf{L}(0)\| = 1$ となる任意の N 次元ベクトルを準備すれば、(6) 式の最大リアプノフ指数は、RNN の出力時系列 $\mathbf{x}(0), \mathbf{x}(1), \dots, \mathbf{x}(T)$ と結合係数 w_{ij} により計算できる。

2.2 Jordan 型 RNN と最大リアプノフ指数

本稿では、外部入力がなく、中間層が1層、出力層の入出力特性が線形であるような Jordan 型 RNN を考える。すなわち、出力層の入出力関係は $i = 1, 2, \dots, N$ として

$$x_i(t+1) = \sum_{k=1}^H v_{ik} z_k(t) + \theta_i, \quad (12)$$

で定義され、中間層の入出力関係は $k = 1, 2, \dots, H$ として

$$z_k(t) = \frac{1}{1 + \exp(-2s_k(t)/s_0)}, \quad (13)$$

$$s_k(t) = \sum_{i=1}^N w_{ki} x_i(t) + \phi_k, \quad (14)$$

で与えられる。ここで、 N , x_i , θ_i は、それぞれ出力層のユニット数、出力、閾値を表し、 H , z_k , s_k , ϕ_k は、それぞれ中間層のユニット数、出力、入力、閾値である。また、 v_{ik} は、中間層ユニット k から出力層ユニット i への結合重み係数、 w_{ki} は、出力層ユニット i から中間層ユニット k へのフィードバック結合の重み係数である。

(12) 式～(14) 式の Jordan 型 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。

$$DF_{ij}(\mathbf{x}(t)) = \frac{2}{s_0} \sum_{k=1}^H v_{ik} \cdot w_{kj} Z_k(t). \quad (15)$$

ただし、 $Z_k(t) = z_k(t)(1 - z_k(t))$ である。したがって、 $L(t+1)$ の要素は次式で表されるから、最大リアプノフ指数は完全結合型と同様に計算できる。

$$L_i(t+1) = \frac{2}{s_0} \sum_{k=1}^H \left(v_{ik} Z_k(t) \sum_{j=1}^N w_{kj} \delta L_j(t) \right). \quad (16)$$

3. 最大リアプノフ指数の制御法

ここでは、あるネットワークの最大リアプノフ指数を所望の値に設定する問題を考える。最大リアプノフ指数は、前節に示したように結合重み w_{ki} , v_{ik} などのネットワークパラメータによって計算される。したがって、ネットワークダイナミクスが目標とする最大リアプノフ指数 λ^{obj} の複雑さをもつように設定するには、適当なネットワークパラメータ値を設定する問題に帰着される。

この場合、ネットワークの学習法のようにパラメータを適当な値で初期化して、降下法などを用いて所望のパラメータ値を得る方法が考えられる。すなわち、ネットワークパラメータを代表して p と書き、ネットワークの最大リアプノフ指数を λ とすると、目標の λ^{obj} にするためのパラメータ p の改変量 δp は、目標値との二乗誤差を $e_\lambda = (\lambda - \lambda^{obj})^2/2$ として

$$\begin{aligned} \delta p &= -\eta \frac{\partial e_\lambda}{\partial p}, \\ &= -\eta (\lambda - \lambda^{obj}) \frac{\partial \lambda}{\partial p}, \end{aligned} \quad (17)$$

で与えられる。ここで、 $\eta (> 0)$ は学習係数である。 $\partial \lambda / \partial p$

は (6) 式より、

$$\frac{\partial \lambda}{\partial p} = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T \frac{1}{L(t)} \frac{\partial L(t)}{\partial p}, \quad (18)$$

であり、 $\partial L(t) / \partial p$ は (11) 式より、

$$\frac{\partial L(t)}{\partial p} = \frac{1}{L(t)} \sum_{i=1}^N L_i(t) \frac{\partial L_i(t)}{\partial p}, \quad (19)$$

である。よって、 $\partial L_i(t) / \partial p$ を求めると改変量 δp が計算できる。以下では、完全結合型 RNN の w_{ij} についての改変量と、Jordan 型 RNN の w_{ki} , v_{ik} についての改変量を求めるが、閾値 θ_i などについても、逆伝搬学習法と同様に、常に1を出力する仮想的なユニットとの結合係数と考えることにより計算できる。

3.1 完全結合型 RNN の最大リアプノフ指数制御

(10) 式より、

$$\begin{aligned} \frac{\partial L_k(t+1)}{\partial w_{ij}} &= \frac{2}{s_0} \left(\frac{\partial X_k(t+1)}{\partial w_{ij}} \sum_{m=1}^N w_{km} \delta L_m(t) \right. \\ &\quad \left. + X_k(t+1) [\Delta_{ik} \delta L_j(t) \right. \\ &\quad \left. + \sum_{m=1}^N w_{km} \frac{\partial \delta L_m(t)}{\partial w_{ij}}] \right). \end{aligned} \quad (20)$$

ここで、 $i = k$ のとき $\Delta_{ik} = 1$ 、それ以外のとき $\Delta_{ik} = 0$ である。 $\partial X_k(t) / \partial w_{ij}$ は定義より

$$\frac{\partial X_k(t)}{\partial w_{ij}} = (1 - 2x_k(t)) \frac{\partial x_k(t)}{\partial w_{ij}}, \quad (21)$$

となり、 $\partial x_k(t) / \partial w_{ij}$ は、BPTT 法、RTRL 法などの従来の RNN の学習アルゴリズム^{12), 13), 16)}により計算できる。また、(5) 式より

$$\frac{\partial \delta L_m(t)}{\partial w_{ij}} = \frac{1}{L(t)} \frac{\partial L_m(t)}{\partial w_{ij}} - \frac{L_m(t)}{L^2(t)} \frac{\partial L(t)}{\partial w_{ij}}, \quad (22)$$

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、(20) 式より $\partial L_k(1) / \partial w_{ij}$ が求まり、(19) 式より $\partial L(1) / \partial w_{ij}$ が計算できる。これらより、(22) 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、 $\partial \lambda / \partial w_{ij}$ が求まる。

3.2 Jordan 型 RNN の最大リアプノフ指数制御

はじめに、 v_{ik} について求める。 $dl_k(t)$ を

$$dl_k(t) = \sum_{i=1}^N w_{ki} \delta L_i(t), \quad (23)$$

で定義すると、(16) 式より、

$$\begin{aligned} \frac{\partial L_j(t+1)}{\partial v_{ik}} &= \frac{2}{s_0} \{ \Delta_{ij} Z_k(t) dl_k(t) \\ &\quad + \sum_{m=1}^H v_{im} \left(\frac{\partial Z_m(t)}{\partial v_{ik}} dl_m(t) \right. \\ &\quad \left. + Z_m(t) \sum_{o=1}^N w_{mo} \frac{\partial \delta L_o(t)}{\partial v_{ik}} \right) \}. \end{aligned} \quad (24)$$

ここで、 $\partial Z_m(t) / \partial v_{ik}$ は

$$\frac{\partial Z_m(t)}{\partial v_{ik}} = (1 - 2z_m(t)) \frac{\partial z_m(t)}{\partial v_{ik}}, \quad (25)$$

である。したがって (12) 式～ (14) 式より

$$\frac{\partial z_m(t)}{\partial v_{ik}} = \frac{2}{s_0} Z_m(t) \sum_{j=1}^N w_{mj} \frac{\partial x_j(t)}{\partial v_{ik}}, \quad (26)$$

$$\frac{\partial x_j(t)}{\partial v_{ik}} = \Delta_{ij} z_k(t-1) + \sum_{o=1}^H v_{jo} \frac{\partial z_o(t-1)}{\partial v_{ik}}, \quad (27)$$

であるから、 $\partial x_j(0)/\partial v_{ik} = 0$, $j = 1, 2, \dots, N$ とおくことによって計算できる。また、 $\partial \delta L_o(t)/\partial v_{ik}$ は

$$\frac{\partial \delta L_o(t)}{\partial v_{ik}} = \frac{1}{L(t)} \frac{\partial L_o(t)}{\partial v_{ik}} - \frac{L_o(t)}{L^2(t)} \frac{\partial L(t)}{\partial v_{ik}}, \quad (28)$$

となるから、完全結合型の (22) 式と同様に計算できる。

つぎに、 w_{ki} について求める。 v_{ik} の場合と同様にして、

$$\begin{aligned} \frac{\partial L_j(t+1)}{\partial w_{ki}} &= \frac{2}{s_0} \sum_{m=1}^H v_{jm} \left(\frac{\partial Z_m(t)}{\partial w_{ki}} dl_m(t) \right. \\ &\quad \left. + Z_m(t) \frac{\partial dl_m(t)}{\partial w_{ki}} \right), \end{aligned} \quad (29)$$

$$\frac{\partial dl_m(t)}{\partial w_{ki}} = \Delta_{km} \delta L_i(t) + \sum_{o=1}^N w_{mo} \frac{\partial \delta L_o(t)}{\partial w_{ki}}, \quad (30)$$

である。ここで、 $\partial Z_m(t)/\partial w_{ki}$ は

$$\begin{aligned} \frac{\partial z_m(t)}{\partial w_{ki}} &= \frac{2}{s_0} Z_m(t) (\Delta_{km} x_i(t) \\ &\quad + \sum_{j=1}^N w_{mj} \frac{\partial x_j(t)}{\partial w_{ki}}), \end{aligned} \quad (31)$$

$$\frac{\partial x_j(t)}{\partial w_{ki}} = \sum_{o=1}^H v_{jo} \frac{\partial z_o(t-1)}{\partial w_{ki}}, \quad (32)$$

より (25) 式の場合と同様にして計算できる。また、 $\partial \delta L_o(t)/\partial w_{ki}$ は (28) 式と同様にして求まる。

4. シミュレーション結果

4.1 完全結合型 RNN の最大リアプノフ指数の設定

$N = 8$ の完全結合型 RNN の重み係数をランダムに初期化した後、提案手法によって最大リアプノフ指数を $\lambda = -2.0$ に設定するシミュレーション結果を Fig. 1 に示す。これより、最大リアプノフ指数の設定に成功していることがわかる。ただし、 $\eta = 0.01$, $T = 3000$ とした。ここで、とくに $\lambda < 0$ の場合、(18) 式をそのまま計算すると $L(t) \approx 0$ のとき、 $\delta p \rightarrow \infty$ となって計算が発散してしまう場合があるため、 $|\delta p|$ の上限を 0.1 に制限した。また、(21) 式を解くために RTRL 法を用いて (18) 式の微係数を計算する場合、計算量 (必要な乗算回数) は $O(N^5 T)$ 、必要なメモリ量は $O(N^3)$ となり、ニューロンの個数 N が増えると膨大な計算量になってしまう。この問題を解決するための近似計算法についてつぎに述べる。

4.2 近似計算による最大リアプノフ指数の設定

はじめに、次式のパラメータ $\overline{\sigma^2}$ を導入する⁸⁾。

$$\overline{\sigma^2} = \frac{1}{12N} \sum_{i,j} w_{ij}^2. \quad (33)$$

Fig. 1 The Lyapunov exponent λ as a function of the iteration by the proposed control method.

これを用いると、最大リアプノフ指数を所望の値に改変するための勾配法に用いる微係数は次式で計算できる。

$$\frac{\partial \lambda}{\partial w_{ij}} = \frac{\partial \lambda}{\partial \overline{\sigma^2}} \cdot \frac{\partial \overline{\sigma^2}}{\partial w_{ij}}. \quad (34)$$

ここで、 $\partial \overline{\sigma^2}/\partial w_{ij}$ は (33) 式より次式で計算される。

$$\frac{\partial \overline{\sigma^2}}{\partial w_{ij}} = \frac{1}{6N} w_{ij}. \quad (35)$$

このとき、 w_{ij} をランダムに設定し、閾値を

$$\theta_i = -\frac{1}{2} \sum_{j=1}^N w_{ij}, \quad (36)$$

で設定すると、 N が大きい RNN における、パラメータ $\overline{\sigma^2}$ と最大リアプノフ指数の近似的な関係より、

$$\frac{\partial \lambda}{\partial \overline{\sigma^2}} \approx \begin{cases} A_1 A_2 \exp(-A_2 \overline{\sigma^2}), & (\lambda < 0), \\ B_1, & (\lambda > 0), \end{cases} \quad (37)$$

となるから^{4), 8)}, (34) 式より近似的な微係数を計算できる。ここで、 A_1, A_2, B_1 は正の定数である。この近似計算法では、 $\overline{\sigma^2}$ を計算するために $O(N^2)$ の計算量が必要になるだけであり、 N が大きなネットワークでも実用的である。この近似計算法によって $N = 100$ の RNN の最大リアプノフ指数を 0 に設定した例を Fig. 2 に示す。これより、初期の最大リアプノフ指数として、 $\lambda < 0$ の場合、 $\lambda \approx 0$ の場合、 $\lambda > 0$ の場合いずれに対してもうまく設定できていることがわかる。この近似法は、閾値に制限を課し、かつ、 N が大きい場合にしか有効ではないが、近似を用いない場合に比べて計算量が極端に少なくなる点が利点である。

4.3 最大リアプノフ指数制御を併用した学習

3.2 で提案した Jordan 型 RNN の最大リアプノフ指数の制御 (Complexity Control: CC) 法を従来の誤差逆伝搬 (Back-Propagation: BP) 学習法と組み合わせることにより、対象システムのダイナミクスの複雑さ (最大リアプノフ指数) も含めて学習する問題を考える。すなわち、対象アトラクタの学習データとの誤差を e_c 、最大リアプノフ指数の誤差を e_λ とし、総合的な誤差 $e = e(e_c, e_\lambda)$ を最小化する。本稿では、簡単のため C を重み係数として $e = e_c + C e_\lambda$ とする。したがって、 e を減少させるパラメータの改変量 δp は次

Fig. 2 The Lyapunov exponent λ as a function of the iteration by the approximation control method.

式で計算する。

$$\delta p = -\eta \left(\frac{\partial e_c}{\partial p} + C \frac{\partial e_\lambda}{\partial p} \right). \quad (38)$$

ここで、第1項は誤差逆伝搬法により計算し、第2項は提案手法により計算する。

以下では、対象システムとして次式のロジスティック関数を用いたシミュレーション結果について考察する。

$$y(t+1) = ay(t)(y(t)-1). \quad (39)$$

ただし、 $a = 3.9$ とした。この場合、対象システムの最大リアプノフ指数は、 $\lambda^{obj} = 0.49$ である。また、このシミュレーションでは、 $H = 4$, $N = 1$, $\eta = 0.0001$, $T = 1000$ とした。この問題において、単に教師信号との誤差を小さくすることだけを目的とすれば、教師信号 y をフィードバックすることにより従来の学習法 (BP) のみによって良い学習結果が得られる (Fig. 3 上図点線: ただし、後述の提案手法の学習結果 (実線) とはほぼ重なっているため、点線は見えない)。しかし、Fig. 3 下図の点線で示すように学習がほぼ終了した後 (700回以降) でも、(12) 式で計算される RNN の出力 x_i をフィードバックした場合の最大リアプノフ指数は、対象システムの最大リアプノフ指数に近いという保証はない。

一方、(38) 式において $C = 1$ とした場合のシミュレーション結果を Fig. 3 (実線) に示す。これより、提案手法との組合せにより (BP+CC)、教師信号との誤差は図において重なって一本に見えるように、BP のみだけの場合とほぼ変わらず、さらに最大リアプノフ指数に関しても、対象システムのそれに近いモデルが得られているのがわかる。すなわち、シミュレーションでは従来法で学習した RNN の最大リアプノフ指数は $\lambda < 0$ であるのに対し、提案手法との組合せにより $\lambda > 0$ となってカオスになっていることがわかる。

5. ま と め

本稿では、RNN のダイナミクスの複雑さの指標の一つである最大リアプノフ指数を所望の値に設定するアルゴリズムを提案した。また、構成ニューロン数が多い場合の計算コストが高くなる問題点を解決するための近似アルゴリズムを示

Fig. 3 The conventional error e_c and the Lyapunov exponent λ as functions of the learning iteration for the Logistic function. The results of learning by the Back-Propagation (BP) algorithm are shown as dotted lines, and solid lines show the results by the proposed method, which is a combination of the BP and our Complexity Control (CC) algorithms.

した。さらに、提案アルゴリズムと従来の学習アルゴリズムの組合せにより、対象ダイナミクスをその時系列学習データの近似だけでなく、対象システムの最大リアプノフ指数も含めた学習が可能であることを示した。

提案手法と従来の学習法を併用した場合、(38) 式の重み係数 C を大きくすると、より早く目的の最大リアプノフ指数に近付くことは可能であるが、従来の学習法のみによるパラメータ変更と併用法の変更量が大きく異なるため教師信号との誤差を減少させる学習速度は、遅くなる傾向が見られた。最適な係数 C の値に関する考察は、今後の課題である。

また、提案手法による最大リアプノフ指数の制御過程より、最大リアプノフ指数は、パラメータ空間上で非常に複雑な関数になっていることが予想され、学習曲線が大きく振動している場合が多いなど、学習係数 η の値などに敏感である。さらに、アルゴリズムの性質上 $|\delta p|$ の上限値を定めなければ計算が発散する可能性があるなどの問題点もある。より効率的な最大リアプノフ指数の制御法構築のためには、パラメータ空間上の関数曲面の大域的な形状解析などが必要である。

参 考 文 献

- 1) 合原 一幸: カオス, サイエンス社 (1990)
- 2) 合原 一幸: カオスセミナー, 海文堂 (1994)
- 3) 船橋 賢一: ニューラル・ネットワークによる連続写像の近似的実現について, 電子情報通信学会技報, MBE88-9, 59/65

- (1988)
- 4) N. Honma, K. Kitagawa, K. Abe and H. Takeda: An Autonomous Criterion of Learning Methods for Recurrent Neural Networks, Proc. of 2nd ASCC, **II**, 219/222 (1997)
 - 5) J. J. Hopfield and D. W. Tank: "Neural" Computation of Decisions in Optimization Problems, Biological Cybernetics **52**, 141/152 (1985)
 - 6) M. Jordan: Generic Constraints on Underspecified Target Trajectories, Proc. International Joint Conference on Neural Networks(IJCNN-89), **I**, 217/225 (1989)
 - 7) 鎌内, 本間, 阿部: 生物的な認識ダイナミクスをもとに構成した文字認識ニューラルネットワークによる未知パターンの認識, 第10回自律分散システム・シンポジウム資料, 13/16 (1998)
 - 8) 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**, 1093/1098 (1997)
 - 9) 中野 馨他: ニューロコンピュータの基礎, コロナ社 (1990)
 - 10) K. S. Narendra and K. Parthasarathy: Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. Neural Networks, **1-1**, 4/27 (1990)
 - 11) 西川, 北村: ニューラルネットと計測制御, 朝倉書店 (1995)
 - 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) D. E. Rumelhart, G. E. Hinton and R. J. Williams: Learning Representations by Back-Propagating Errors, Nature, **323**, 533/536 (1986)
 - 15) C. A. Skarda and W. J. Freeman: How brains make chaos in order to make sense of the world, Behavioral and Brain Sciences, **10**, 161/195 (1987)
 - 16) R. J. Williams and D. Zipser: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, Neural Computation, **1-2**, 270/280 (1989)
 - 17) A. Wolf, J. B. Swift, H. L. Swinney and J. A. Vastano: Determining Lyapunov Exponents from a time series, Physica, **16D**, 285/315 (1989)

[著者紹介]

本間 経 康 (正会員)

1995年, 東北大学大学院工学研究科博士後期課程修了。同年同大学電気通信研究所研究員。96年同大学医療技術短期大学部講師。98年同助教授。最適制御, 自律分散システム, ニューラルネットワーク, 人工生命などの研究に従事。IEEE, 日本放射線技術学会の会員(工学博士)。

酒 井 正 夫 (学生会員)

1997年, 東北大学工学部電子工学科卒業。現在同大学大学院工学研究科博士前期課程に在学中。リカレントニューラルネットワークの学習に関する研究に従事。

阿 部 健 一 (正会員)

1969年, 東北大学大学院工学研究科博士課程修了。同年同大学工学部助手。74年同助教授。84年豊橋技術科学大学教授。93年東北大学教授。デジタル制御, 適応・学習制御などの研究に従事。IEEE, 電気学会, 情報処理学会などの会員(工学博士)。

竹 田 宏 (正会員)

1959年, 東北大学大学院工学研究科博士課程修了, 同年富士通(株)入社。60年東北大学電気通信研究所助手。62年同大学助教授。73年同教授。93年東北学院大学教授。最適推定理論, 手動制御系の数学的モデルの研究などに従事。IEEE, 電気学会, 電子情報通信学会などの会員, 本学会フェロー, 名誉会員(工学博士)。