

# 自己構成型大脳新皮質学習における時間遅れシナプスの検討

○鈴木嶺聡哲 青木健 高玉圭樹 佐藤寛之 (電気通信大学)

## Time Delay Synapses in Self-Structured Cortical Learning Algorithm

\*S. Suzugamine and T. Aoki and K. Takadama and H. Sato  
(The University of Electro-Communications)

**Abstract**— The cortical learning algorithm is one of time-series data prediction algorithms based on the human neuro cortex. The cortical learning algorithm uses multiple columns to represent input data, and each column has multiple cells to represent time-series pattern of input data. We previously proposed a self-structured cortical learning algorithm which dynamically adding columns and cells according to input data. In this work, We apply time delay synapses to a self-structured cortical learning algorithm and compare the prediction accuracy between conventional and proposed method. Although conventional method predicted using only input data at one step ago, proposed method predict using it in a certain past section. Experimental results using test time-series data involving a sine wave, its combined one, and the logistic mapping show the prediction accuracy is improved.

**Key Words:** time-series data prediction, cortical learning algorithms, hierarchical temporal memory

### 1 はじめに

時系列データを予測する階層時間記憶は、哺乳類の大脳新皮質をモデル化した概念である<sup>1)</sup>。階層時間記憶を具現化したアルゴリズムの一つに大脳新皮質学習アルゴリズム (Cortical Learning Algorithm, 以下 CLA)<sup>2, 3)</sup>がある。CLA の学習器は、複数のカラムと、複数のセルによって構成される。各カラムは、複数のセルを内包する。カラムは、入力データビットとのシナプス接続によって活性状態になり、複数のカラムの活性パターンによって入力データを表現する<sup>4, 5)</sup>。セルは、セル間のシナプス接続によって活性状態になり、複数のセルの活性パターンによって入力データの時系列パターンを表現・予測する。関連研究として、従来のニューラルネットワークに基づく時系列予測モデルであるリカレントニューラルネットワーク<sup>6)</sup>や長短期記憶 (Long Short-Term Memory, 以下 LSTM)<sup>7)</sup>がある。これらは、シナプスネットワークを構築するために、計算コストがかかる最適化を要する。一方、CLA は、シナプスネットワークを簡素なアルゴリズムで構築できるところに利点がある。応用例としては、異常検知への適用例<sup>8)</sup>がある。また、タクシー乗車数の時系列予測において、CLA は LSTM より高い予測精度を示すことが報告されている<sup>9)</sup>。

CLA は、カラムとセルの数をパラメータとして事前に設定しなければならないところに問題がある。適切なカラムとセルの数は入力データによって異なるが、入力データは未知であるため、適切なカラムとセルの数を事前に設定することは困難である。これに対して、これまでに我々は、入力データに対してオンラインでカラムとセルを自己構成する自己構成型 CLA を提案した<sup>10)</sup>。しかし、自己構成型 CLA は、1 時点前の入力データのみを使用して予測をするため、長期的な文脈依存の時系列タスクへの対応や、予測誤差の収束速度に課題がある。

本稿では、複数時点前の入力データに基づいて予測する時間遅れシナプスを提案する。これにより、過去のある一定の区間の入力データから予測が可能となり、長期的な文脈依存の時系列タスクへの対応や収束速度

の向上が見込まれる。実験では、自己構成型 CLA に時間遅れシナプスを適用した場合の時系列データの予測精度を検証する。時系列テストデータとして正弦波、正弦波の合成波、ロジスティック写像を用いて予測精度が改善することを示す。

### 2 大脳新皮質学習アルゴリズム

#### 2.1 構成

CLA の学習器の構成を Fig. 1 に示す。CLA の学習器全体をリージョンと呼ぶ。リージョン内には、複数のカラムが存在する。それぞれのカラム内には、複数のセルが存在する。入力データはバイナリビット列で表現し、各ビットとカラムの間にシナプスを用意する。また、セルとセルの間にもシナプスを用意する。Fig. 1 は、リージョン内に 5 本のカラムがあり、各カラムが 5 個のセルを持つ場合の例である。

##### 2.1.1 カラム

カラムには、活性と通常の 2 状態がある。各カラムは、シナプス接続された入力データビットの 1 の数が一定数以上のときに活性状態になり、それ以外は通常状態になる。カラムの役目は、入力データビット列をカラム群の活性パターンによって学習器の内部表現に変換することである。カラム数が多い場合は内部表現できる入力データの粒度が高く、カラム数が少ない場合は内部表現できる入力データの粒度が低くなる。また、過剰なカラムは予測精度の悪化と計算時間の増加

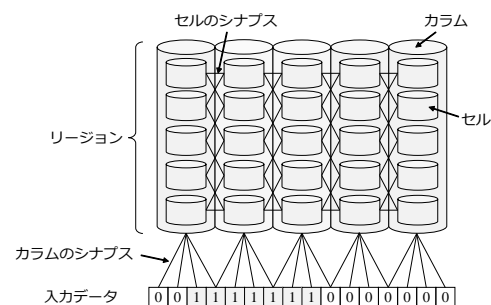


Fig. 1: CLA の構成

につながる。

従来の CLA では、カラム数を事前にパラメータとして決定する必要がある。

### 2.1.2 セル

セルには、通常、活性、予測の 3 状態がある。これらの状態を用い、セルは二つの役目をこなす。

一つ目は、同一のデータ値が入力され、カラムによる内部表現が等しくても、その前に入力されたデータの時系列が異なれば、それを区別できるようにする役目である。これには、セルの活性状態が使われる。具体的には、活性したカラムの中に、予測状態のセルが存在すれば、そのセルを活性させる。予測状態のセルが存在しない場合、すべてのセルを活性させる。このように、入力されたデータの文脈情報を表現する。

二つ目は、次の入力データを予測する役目である。これには、セルの予測状態が使われる。具体的には、すべてのセルについて、シナプス接続されたセルの活性数が一定数以上のセルを予測状態にする。すなわち、次の入力データで活性状態になるカラム内のセルを予測状態にできれば、予測成功ということになる。セル数は多いほど、多数の時系列パターンを表現可能になる。また、過剰なセルは予測精度の悪化と計算時間の増加につながる。しかし、適切なセル数は入力データによって異なる。

従来の CLA では、セル数を事前にパラメータとして決定する必要がある。また、すべてのカラムにおいて等しいセル数が用いられる。

## 2.2 入力表現

実数値のデータをバイナリビット列に変換して CLA の学習器に入力する。本研究では、時点  $t$  において入力される実数値  $X(t)$  を、バイナリビット列における 1 のチャンクの位置に変換する。実数値の入力データの最大値を  $max$ 、最小値を  $min$ 、変換後のバイナリビット列の長さを  $n$ 、チャンクの長さを  $w$  とすると、入力バイナリビット列  $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t)) \in \{0, 1\}^n$  は、次式で表される。

$$x_i(t) = \begin{cases} 1, & \text{If } \left\lfloor \frac{(X(t)-min) \cdot (n-w)}{max-min} \right\rfloor < i \leq \left\lfloor \frac{(X(t)-min) \cdot (n-w)}{max-min} \right\rfloor + w, \\ 0, & \text{Otherwise,} \end{cases}$$

( $i = 1, 2, \dots, n$ ). (1)

## 2.3 空間プーリング・時間プーリング

まず、空間プーリングでは、 $t$  時点における入力データ  $\mathbf{x}(t)$  を、カラムの活性パターンによって学習器の内部表現に変換する<sup>4, 5)</sup>。

次に、空間プーリングによって決定した活性状態のカラム内のセルのうち、活性状態にするセルを決定する処理を時間プーリングという。これにより、入力データの時系列における出現パターンを学習器内において表現する。さらに、現在の活性状態のセルから、次の入力データで活性状態になるセルを予測するために、セルを予測状態にする処理を施す。これにより、時系列におけるデータの出現パターンを予測する。

## 2.4 予測値

予測状態のセル群から予測値を算出する。本稿では、まず、各入力データビットに対して、予測状態のセルが存在するカラムと接続状態にあるシナプス数を計数する。その数が大きい順に入力データビットのチャンク長  $w$  までを 1、それ以外を 0 として予測データビット列を生成する。予測データビット列から実数の予測値へ変換するとき、全ビット長  $n$  における 1 のチャンク長  $w$  で表現できるすべてのビット列に対して、予測データビット列とのハミング距離が最も短いビット列に対応する実数値を出力する。

## 3 従来法：自己構成型 CLA<sup>10)</sup>

従来の CLA は、カラムとセルの数を事前にパラメータとして設定しなければならない。カラムは多いほど、入力データを高精度に内部表現へ変換できる。セルは多いほど、時系列における出現パターンを高精度に内部表現へ変換できる。しかし、カラムとセルは、多いほど計算コストを必要とするだけでなく、学習速度も低下するため、単に多くすれば良いというわけにはいかない。また、入力する時系列データによって、適切なカラムとセルの数は異なる。入力データは未知であるため、適切なカラムとセルの数を事前に設定することは困難である。さらに、従来の CLA は、すべてのカラムが同一数のセルを持つが、入力データによっては、多数のセルを必要とするカラムとそうでないカラムが生じる。そのため、セル数の設定によっては、セルが不足するカラムが存在したり、無駄なセルが存在するカラムが存在したりする。

これらの問題を解決するために、我々は入力データに合わせてカラムとセルの数を自己調整し、学習器を自己構成する自己構成型 CLA を提案した<sup>10)</sup>。これにより、カラムとセル数の事前設定を回避可能にした。自己構成型 CLA には、カラムを動的構成する動的空間プーリングと、セルを動的構成する動的時間プーリングが導入されている。

### 3.1 動的空間プーリング

動的空間プーリングでは、入力データに合わせてカラムを構成した上で、入力データをカラムの活性パターンによる内部表現に変換する。

まず、存在する各カラムについて、データビット 1 と接続状態にあるシナプス数を重複数  $ol_c$  として計数し、閾値  $T_{ol_c}$  以上のカラムを活性候補にする。次に、各入力データビットについて、活性候補のカラムと接続状態にあるシナプス数を重複数  $ol_i$  として計数する。活性候補のカラム数が  $N^{ac}$  より少ない場合、リージョン内にカラムを一つ追加する。追加したカラムのシナプスは、重複数  $ol_i$  が少ない入力データビット 1 から、順に  $N^s$  本設置する。ただし、 $N^s \geq T_{ol_c}$  に設定する。結果として、追加したカラムは活性候補になる。この追加処理を、活性候補のカラム数が  $N^{ac}$  になるまで繰り返す。次に、活性候補のカラムから、カラム数  $N^{ac}$  までを活性状態にする。上述の処理でカラムを追加した場合は、活性候補にしたすべてのカラムを活性状態にすることになる。カラムを追加しておらず、活性候補のカラム数が  $N^{ac}$  より多い場合は、重複数  $ol_c$  が多い順に  $N^{ac}$  本のカラムを活性状態にする。

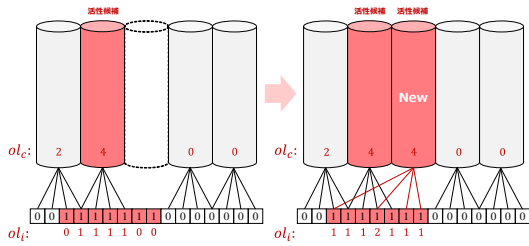


Fig. 2: 自己構成型 CLA<sup>10)</sup>: 動的空間プーリング

Fig. 2 は、活性状態にするカラム数  $N^{ac} = 2$  に対して、活性候補になるカラムが1つしかなく、もう1つのカラムを追加する例を示している。追加されたカラムは、活性候補になったカラムとのシナプス接続数  $ol_i$  が小さい入力データビットの順に、赤色のシナプスを配置することがわかる。

### 3.2 動的時間プーリング

動的時間プーリングは、活性状態のカラムの集合に対して、セルを動的構成しながら活性状態のセル、さらに予測状態のセルを決定する処理である。

まず、活性状態のカラム内に予測状態のセルが存在する場合は、そのセルを活性状態にする。活性状態のカラム内に予測状態のセルが存在しない場合は、確率  $p_{rep}$  でリプレイ処理を実行する。

#### 3.2.1 リプレイ

リプレイは、過去  $\delta$  時点をさかのぼって活性状態になったセルのシナプスを切断し、新たにセルを再生成することでシナプスネットワークを再構築する修復処理である。具体的には、時間を  $\delta$  時点さかのぼり、 $t - \delta$  時点において活性状態のセルと、 $t - \delta + 1$  時点において活性状態のセルとのシナプスを切断する。 $t - \delta + 1$  時点において活性状態のカラム内に新しいセルを生成し、既存の活性状態のセルの代わりに活性状態にする。生成したセルと、そのときに活性状態である他のセルとの間に新しいシナプスを構築する。この処理を  $t - 1$  時点まで繰り返す。最後に、 $t$  時点で活性状態のカラム内に新しくセルを生成し、そのセルと  $t - 1$  時点において活性状態のセルとの間にシナプスを構築する。新しく生成したセル群を  $t$  時点で活性したセルとする。

Fig. 3 にリプレイ処理の例を示す。 $t$  時点において、活性状態ではあるが予測セルが存在しないひとつのカラムに注目し、リプレイ処理を施す。さかのぼる過去の時点数を  $\delta = 3$  とし、 $t - 3$ 、 $t - 2$ 、 $t - 1$  と付記したセルは、その時点で活性状態だったことを表す。まず、状態 (a) において、シナプス接続された  $t - 3$ 、 $t - 2$ 、 $t - 1$  時点で活性状態だったセルが存在することがわかる。次の状態 (b) において、それらのシナプスを切断する。状態 (c) において、 $t - 3$ 、 $t - 2$ 、 $t - 1$  時点で活性状態だったセルと同じカラム内に新しく3つセルを生成し、時系列の順番にシナプスを接続する。状態 (d) において、注目するカラム内に新しくセルを生成し、 $t - 1$  時点で活性したセルとシナプスを接続する。このように、リプレイは過去にさかのぼってセルとシナプスネットワークを再構成する。

#### 3.2.2 バックトラック

確率  $1 - p_{rep}$  でリプレイ処理を実行しない場合について、活性状態のカラム内に予測状態のセルが存在しなければ、バックトラック処理を実行する。

バックトラックは、学習器内に新たなセルを生成せずに、過去にさかのぼって活性状態にするセルを決定する処理である。バックトラック処理では、まず、予測状態のセルが存在しない活性状態のカラム内の各セルについて、 $t - 1$  時点で活性状態だったカラム内のセルとのシナプス数を重複数  $ol_e$  として計数し、閾値  $N^{ac}$  以上の場合に活性状態にする。また、 $t - 1$  時点で活性状態だったカラムにおける  $t - 1$  時点の活性状態のセルへシナプスを接続しなおす。

#### 3.2.3 セルの追加

予測状態のセルが存在しない活性状態のカラム内に新たにセルを生成し、 $t - 1$  時点で活性状態だったセルとの間にシナプスを配置、生成したセルを活性状態にする。

#### 3.2.4 セルの削除

自己構成型 CLA では、過剰に生成され不要となったセルを削除する。それぞれのセルは *lifespan* という値を持つ。セルが生成されたとき、その *lifespan* は事前に決定した  $lifespan_0$  に初期化される。それぞれの時点  $t$  で、活性となったセルの *lifespan* の値は  $lifespan_0$  にし、すべてのセルの *lifespan* の値を1減少させる。セルの *lifespan* が0となったときにそのセルは削除される。

この処理によって、過剰に生成された不要なセルを削除する。

#### 3.2.5 予測セルの決定

最後に予測状態にするセルを決定する。各セルについて、活性状態のセルに加えて  $t$  時点でセルが追加されたカラム内のすべてのセルに対して、接続されたシナプス数を  $ol_e$  として計数し、各カラムに一つ以下で  $ol_e$  が大きい順に  $N^{ac}$  個のセルを選択して予測状態にする。

## 4 問題点

従来の自己構成型 CLA における、動的時間プーリングのシナプスは1時点前の入力データのみを使用することにより、長期的な文脈依存の時系列タスクに対応できない問題がある。その問題に対処するために、新たにセルを再生成することで、シナプスネットワークを再構築する修復処理である、リプレイ処理を実行するが、確率的挙動であるため、収束速度にばらつきが生じる。このように従来法には、長期的な文脈依存の時系列タスクに対応できない問題や収束速度の問題が存在する。

## 5 提案法: 時間遅れシナプスを適用した自己構成型 CLA

本稿では、長期的な文脈依存への対応や収束速度を向上させることを目的として、セルに時間遅れシナプスを適用した。提案法における動的時間プーリングは以下の手順となる。

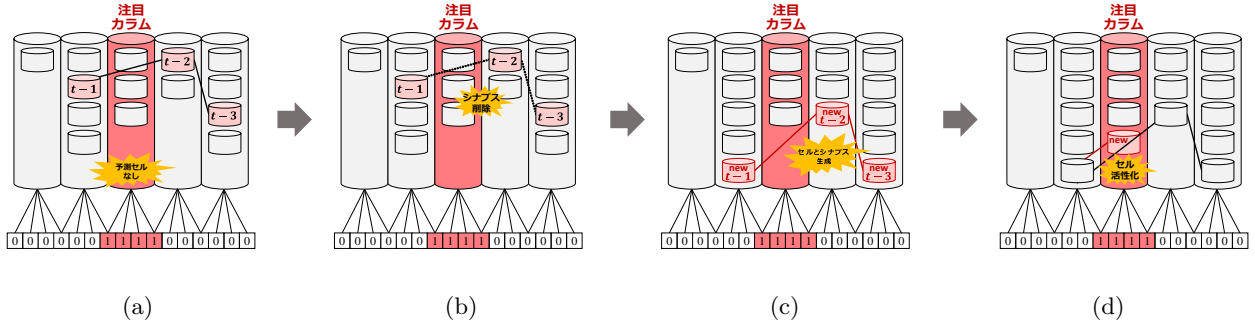


Fig. 3: 自己構成型 CLA<sup>10</sup>: 動的時間プリーングにおけるリプレイ

## 5.1 動的時間プリーング

動的時間プリーングは、活性状態のカラムの集合に対して、セルを動的構成しながら活性状態のセル、さらに予測状態のセルを決定する処理である。

まず、活性状態のカラム内に予測状態のセルが存在する場合は、そのセルを活性状態にする。活性状態のカラム内に予測状態のセルが存在しない場合は、バックトラックを実行する。

### 5.1.1 バックトラック

バックトラックは、学習器内に新たなセルを生成せずに、過去にさかのぼって活性状態にするセルを決定する処理である。バックトラック処理では、まず、予測状態のセルが存在しない活性状態のカラム内の各セルについて、区間  $[t-W, t-1]$  時点で活性状態だったカラム内のセルとの時間遅れシナプス数を重複数  $ol_e$  として計数し、閾値  $S$  以上の場合に活性状態にする。次に、区間  $[t-W, t-1]$  時点で活性状態だったセル群からランダムに  $S$  個選択し、時間遅れシナプスを接続しなおす。

### 5.1.2 セルの追加

予測状態のセルが存在しない活性状態のカラム内に新たにセルを生成し、窓サイズ  $W$  として区間  $[t-W, t-1]$  時点で活性状態だったセルとの間に、シナプス数  $S$  だけ重複しないように、ランダムに時間遅れシナプスを配置する。次に、生成したセルを活性状態にする。

### 5.1.3 セルの削除

従来法と同様の処理をする。

### 5.1.4 予測セルの決定

最後に予測状態にするセルを決定する。各セルについて、区間  $[t-W+1, t]$  時点で、活性状態のセルに加えてセルが追加されたカラム内のすべてのセルに対して、接続された時間遅れシナプス数を  $ol_e$  として計数し、各カラムに一つ以下で  $ol_e$  が大きい順に  $N^{ac}$  個のセルを選択して予測状態にする。

Fig. 4 は、 $W = 3$  とした予測セルを決定する場合の例である。区間  $[t-2, t]$  内のそれぞれ  $t, t-1, t-2$  時点の活性状態であったセルから、各時間遅れのシナプスを使って予測セルが決定される。

## 5.2 期待される効果

まず、時間遅れシナプスを適用することにより、過去のある一定の区間の入力データから予測が可能となり、長期的な文脈依存の時系列タスクへの対応が挙げ

られる。次に、確率的な挙動をするリプレイを実行しないので収束速度の向上が見込まれる。

## 6 実験内容

### 6.1 概要

テスト時系列データである正弦波やその合成波、ロジスティック写像を予測する実験において、自己構成型 CLA が、従来の CLA<sup>9)</sup> や LSTM<sup>7)</sup> より良好な予測性能を示すことをこれまでに報告した<sup>10)</sup>。本稿では従来の自己構成型 CLA と、時間遅れシナプスを適用した提案する自己構成型 CLA の性能を比較する。

### 6.2 時系列入力データ

テスト関数は、先の報告<sup>10)</sup> と同じく、正弦波、合成波、ロジスティック写像を用いた。これらは次式によって定義される。

$$X_{\sin}(t) = \sin\left(\frac{(t-1) \cdot \pi}{50}\right), \quad (2)$$

$$X_{\text{com}}(t) = \sum_{k \in \{1, 3, 5, 7, 9\}} \frac{1}{k} \cdot \sin\left(\frac{(t-1) \cdot k \cdot \pi}{50}\right), \quad (3)$$

$$X_{\log}(t) = \alpha \cdot X_{\log}(t-1) \cdot \{1 - X_{\log}(t-1)\}. \quad (4)$$

ロジスティック写像はパラメータを  $\alpha = \{3.569, 3.7\}$  として入力する。前者は周期的なふるまいをし、後者はカオスなふるまいをする。

### 6.3 パラメータ

Eq. (2)-(4) で定義したテスト関数の実数値を、本実験では、2048 ビットのバイナリ値  $n = 2048, \mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$  に変換する。チャンクサイズは  $w = 40$  とする。Eq. (2) で定義される正弦波と Eq. (3) で定義される合成波では、 $[\min, \max] = [-1.0, 1.0]$  に設定する。二つのロジスティック写像については、 $[\min, \max] = [0.0, 1.0]$  として、初期値は  $X_{\log}(1) = 0.1$  とする。入力時点は  $t \in [1, 10^5]$  とする。

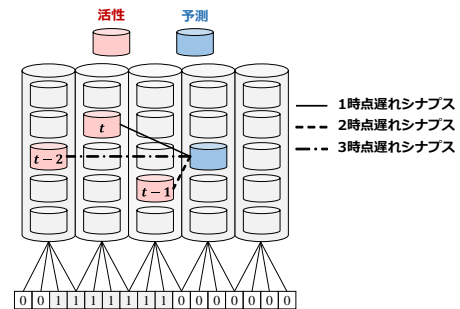


Fig. 4: 時間遅れシナプスによる予測セルの決定

**Table 1:** 総予測誤差

	従来法	提案法
正弦波	64.01	55.70
合成波	234.14	59.39
ロジスティック写像 ( $\alpha = 3.569$ )	34.16	31.58
ロジスティック写像 ( $\alpha = 3.7$ )	258.37	237.46

動的空間プーリングにおけるそれぞれのカラムのシナプスを  $N^s = 2$  とする。また、カラムの活性数を  $N^{ac} = 20$  とする。それぞれのセルの lifespan の初期値を  $lifespan_0 = 5,000$  とする。従来法ではリプレイ確率を  $p_{rep} = 0.003$  とし、リプレイにおいて過去にさかのぼる時点数を  $\delta = 5$  とする。提案法では窓サイズを  $W = 5$  とし、シナプス数を  $S = 20$  とする。

評価尺度として、真値と予測値の絶対誤差である予測誤差を用いる。予測誤差は、小さいほど良好な結果と判断する。各予測アルゴリズムは、独立した 100 回試行の予測誤差の平均値で比較する。

## 7 実験結果と考察

### 7.1 予測誤差

各時系列入力データに対する予測誤差の推移の結果を Fig. 5 に示す。Fig. 6 は、予測誤差の推移の結果を拡大した図である。すべての時点の予測誤差を合計した総予測誤差を Table 1 に示す。

まず、周期関数である正弦波、合成波、ロジスティック写像 ( $\alpha = 3.569$ ) においては、提案法は予測誤差がある一定値に収束しており、その値も従来法を下回っていることがわかる。また、提案法は従来法に比べて予測誤差の減少が早いことが結果からわかる。次に、非周期関数であるロジスティック写像 ( $\alpha = 3.7$ ) では、提案法が従来法よりも予測誤差が小さく推移していることがわかる。さらに、四つのすべての関数において、提案法が従来法に比べて総予測誤差が小さいことがわかる。このように、提案法の予測誤差が従来法よりも小さいことから、次のようなことが考察できる。長期的な文脈依存に対処するために、従来法では、シナプスネットワークを再構築する修復処理である、リプレイという処理を確率的に実行していた。そのため予測性能にばらつきが生じた。一方、提案法ではある一定区間の入力データの情報を用いたことにより従来法に比べて安定した予測が可能となったと考えられる。

これらの結果から、時間遅れシナプスを適用した提案法は、従来法と比べて高い予測性能を示すことが明らかになった。

### 7.2 セル数の推移

各時系列入力データに対する総セル数の推移を Fig. 7 に示す。Fig. 8 は、総セル数の推移の結果を拡大した図である。

この結果から、従来法と提案法は、共に総セル数が入力時点  $t$  の経過に伴って一定値に収束することがわかる。また、その収束値は入力データによって異なることがわかる。その中で、周期関数である正弦波、合成波、ロジスティック写像 ( $\alpha = 3.569$ ) においては、提案法における総セル数の収束値が従来法よりも小さいことがわかる。これは CLA は複数の活性状態のセル

によって、ある時系列の入力データを表現することから、時間遅れシナプスを適用することにより、複数の時点の入力データに対応するセルが増えることによって、全体として必要な総セル数が減少したことが考えられる。以上より、周期関数において、提案法は最適な規模の学習器を構築することに貢献したことがわかる。次に、非周期関数であるロジスティック写像 ( $\alpha = 3.7$ ) では、提案法における総セル数の収束値が従来法よりも大きいことがわかる。これは非周期関数は複雑な時系列データなので、多くのセルが必要となるが、従来法では過分にセルが削除される一方、提案法ではより多くのセルが保持され、活用されていると考えられる。

これらの結果から、時間遅れシナプスを適用した提案法は、従来法と比べて適切なセル数が調整可能であることが明らかになった。

## 8 まとめ

本稿では、長期的な文脈依存への対応や収束速度を向上させることを目的として、自己構成型 CLA に時間遅れシナプスを適用することを提案した。その結果、提案法が従来法に比べて収束速度、予測精度が共に向上することを示した。加えて提案法では、総セル数が一定数に収束することを示した。本稿の実験で用いた周期関数において、総セル数は従来の収束値よりも小さく、より軽量の学習器となった。また、本稿の実験で用いた非周期関数において、複雑な時系列を認識するために、総セル数が従来の収束値よりも大きくなり、セルの不足を解消する結果が得られた。

今後は、不確実な環境における時系列予測問題に取り組む予定である。

## 参考文献

- 1) J. Hawkins, S. Blakeslee: *On Intelligence: How a New Understanding of the Brain Will Lead to the Creation of Truly Intelligent Machines*, Times Books, 2005.
- 2) J. Hawkins, A. Subutai, and D. Dubinsky: "Hierarchical temporal memory including HTM cortical learning algorithms," Technical report, Numenta, Inc, 2010.
- 3) J. Hawkins and A. Subutai: "Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex," *Frontiers in Neural Circuits*, pp. 1–13, 2016.
- 4) Y. Cui and S. Ahmad, J. Hawkins: "The HTM Spatial Pooler—A Neocortical Algorithm for Online Sparse Distributed Coding," *Frontiers in Neuroscience*, doi: 10.3389/fncom.2017.00111, 2017.
- 5) J. Hawkins and S. Ahmad, Y. Cui: "A Theory of How Columns in the Neocortex Enable Learning the Structure of the World," *Frontiers in Neural Circuits*, Vol. 11, Article 81, pp. 1–18, 2017.
- 6) J.L. Elman: "Finding structure in time," *Cognitive Science*, No. 14, pp. 179–211, 1990.
- 7) S. Hochreiter and J. Schmidhuber: "Long Short-Term Memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.
- 8) A. Lavin and S. Ahmad: "Evaluating Real-time Anomaly Detection Algorithms the Numenta Anomaly Benchmark," *Proc. of the 14th International Conference on Machine Learning and Applications (IEEE ICMLA '15)*, pp. 38–44, 2015.
- 9) Y. Cui, S. Ahmad, and J. Hawkins: "Continuous Online Sequence Learning with an Unsupervised Neural Network Model," *Neural Computation*, Vol. 28, Issue. 11, pp. 2474–2504, 2016.



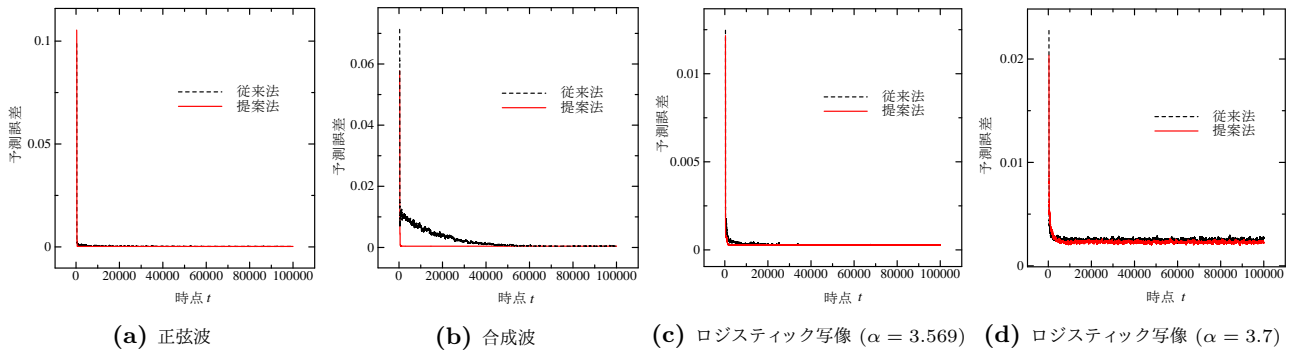


Fig. 5: 予測誤差の推移

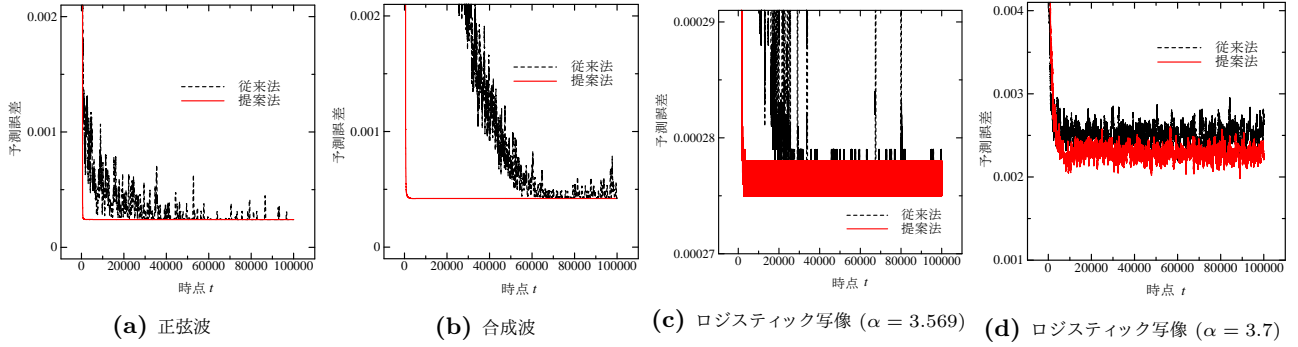


Fig. 6: 予測誤差の推移 (拡大)

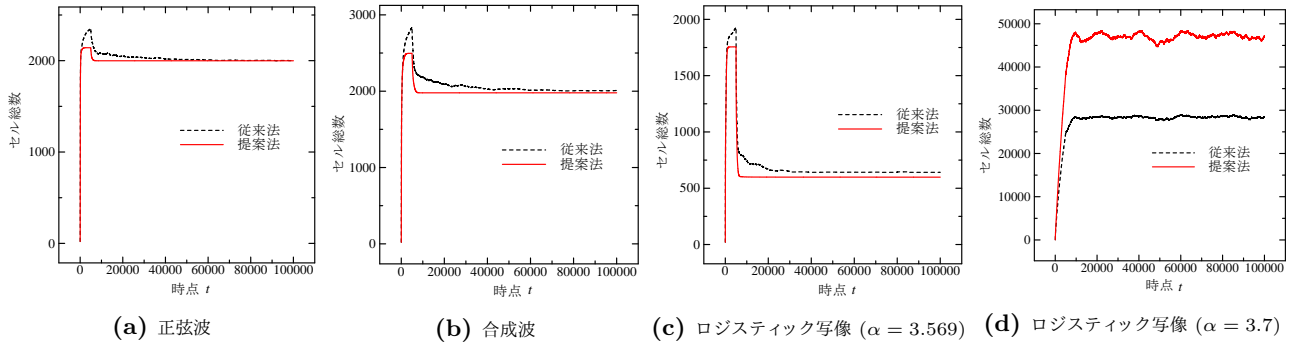


Fig. 7: セル数の推移

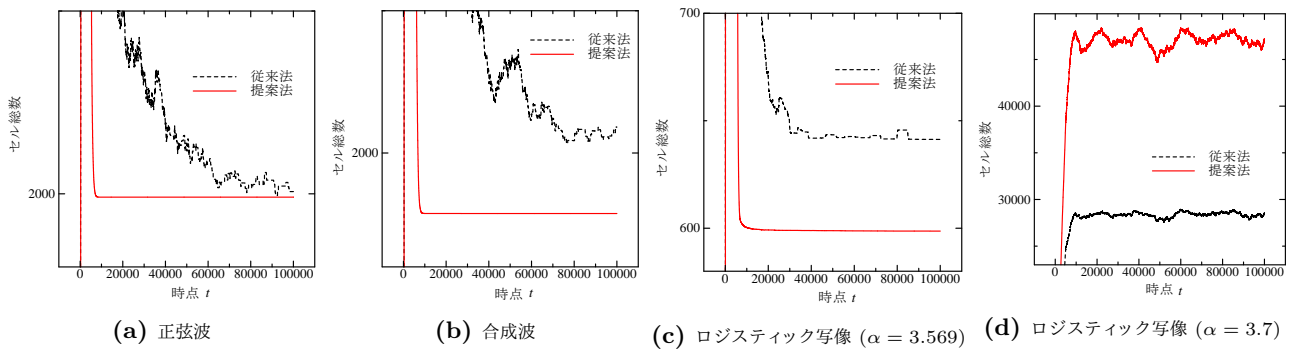


Fig. 8: セル数の推移 (拡大)

- 10) S. Suzugamine, T. Aoki, K. Takadama, and H. Sato, "A Study on a Cortical Learning Algorithm Dynamically Adjusting Columns and Cells," *Proc. of Joint 10th International Conference on Soft Com-*

*puting and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems (SCIS&ISIS2018)*, pp.478–484, 2018.