

大脳新皮質学習における多層化に関する検討

○青木 健 高玉 圭樹 佐藤 寛之 (電気通信大学)

A Study of Multi-layer Structure in Cortical Learning Algorithm

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

Abstract— This work proposes a multi-layered cortical learning algorithm. Cortical learning is a time-series data prediction algorithm designed based on the human neuro-cortex. The human neuro-cortex has a multi-layered structure, while the conventional cortical learning algorithm has a single layer. This work introduces a multi-layered structure into the cortical learning algorithm. The experimental results show that the proposed multi-layered cortical learning algorithm achieves higher prediction accuracy than the single-layered cortical learning algorithm on several artificial time-series data.

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

1 はじめに

人間の脳から発想を得た、階層時間記憶 (Hierarchical Temporal Memory, 以下 HTM)^{1, 2, 3)} という概念がある。HTM に基づく時系列予測手法として、大脳新皮質学習アルゴリズム (Cortical Learning Algorithm, 以下 CLA)^{4, 5, 6)} がある。関連研究として、人工ニューラルネットワークの一種であり、リカレントニューラルネットワーク^{7, 8)} を拡張した長短期記憶 (long short-term memory, 以下 LSTM)⁹⁾ による時系列予測の有用性が知られている。タクシーの需要予測問題¹⁰⁾ において、CLA が LSTM より高い予測精度を達成した事例があり、CLA は有望な時系列予測法のひとつといえる。

CLA は、予測器を複数のカラムで構成する。各時間ステップにおいて、CLA は一定数のカラムを活性状態にすることで、入力データを内部表現にエンコードする。この操作を空間プーリングという。空間プーリングでは、入力データビット列とカラム間のシナプスを使用および学習する。また、各カラムは一定数のセルを持つ。CLA は、一定数のセルを活性状態にして、入力データの文脈を内部文脈表現にする。さらに、CLA はセルを予測状態にして、次の時間ステップの入力データを内部予測表現にする。

従来の CLA は、単一層で構成されている。一方、人間の脳新皮質の構造は多層構造を有することが知られている。予測器を多層化することで、人間のように高次の予測を実現できる可能性がある。これにより、予測精度の向上、および学習速度の向上が期待できる。

本稿では、CLA に多層構造を導入する。第一層は入力データを受け取り、その文脈表現を第二層へ渡す。第二層では渡された文脈情報から高次元の予測を行い、その結果を第一層へフィードバックする。そのフィードバック情報を用いて、第一層における学習を促進することで予測精度を向上する。提案する方法の効果を検証するため、複数の人工時系列データを予測する問題において、従来の CLA と提案法を比較する。

2 大脳新皮質学習アルゴリズム (CLA)

2.1 予測器

CLA^{4, 5, 6)} の予測器の構成を Fig. 1 に示す。CLA は、入力データビット列 $\mathbf{x} = (x_1, x_2, \dots, x_n)$ を各時点

t の入力データとして扱う。CLA の予測器は、 n_c 個のカラム c_i ($i = 1, 2, \dots, n_c$) によって構成される。Fig. 1 では、 $n_c = 5$ の例を示している。

各カラム c_i は、通常と活性の 2 状態のいずれかになる。また、各カラム c_i は n_{cy} 本のシナプス $c_i.y_k$ ($k = 1, 2, \dots, n_{cy}$) を持ち、入力データビットとの関係を形成する。各シナプス $c_i.y_k$ は、接続か切断かを決定する永続値 $c_i.y_k.p$ を持つ。 $c_i.y_k.p \geq \theta_c$ のときにシナプス $c_i.y_k$ は接続され、 $c_i.y_k.p < \theta_c$ のときに切断される。ここで、 θ_c はカラムのシナプスの閾値である。

各カラム c_i には n_r 個のセル $r_{i,j}$ ($j = 1, 2, \dots, n_r$) が内在する。Fig. 1 では、 $n_r = 5$ の例を示している。各セル $r_{i,j}$ は、通常と活性と予測の 3 状態のいずれかになる。また、各セル $r_{i,j}$ はシナプス $r_{i,j}.y_k$ ($k = 1, 2, \dots$) を持ち、他のセルとの関係を形成する。各シナプス $r_{i,j}.y_k$ も、永続値 $r_{i,j}.y_k.p$ を持つ。 $r_{i,j}.y_k.p \geq \theta_r$ のときにシナプス $r_{i,j}.y_k$ は接続され、 $r_{i,j}.y_k.p < \theta_r$ のときに切断される。ここで、 θ_r はセルのシナプスの閾値である。

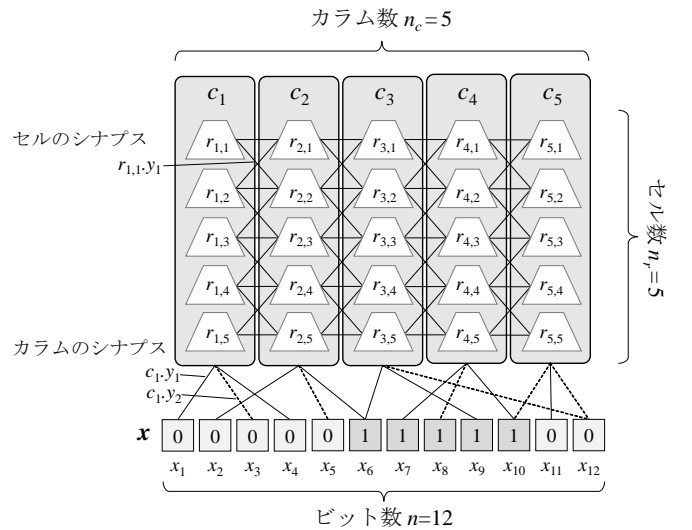


Fig. 1: CLA の構成要素

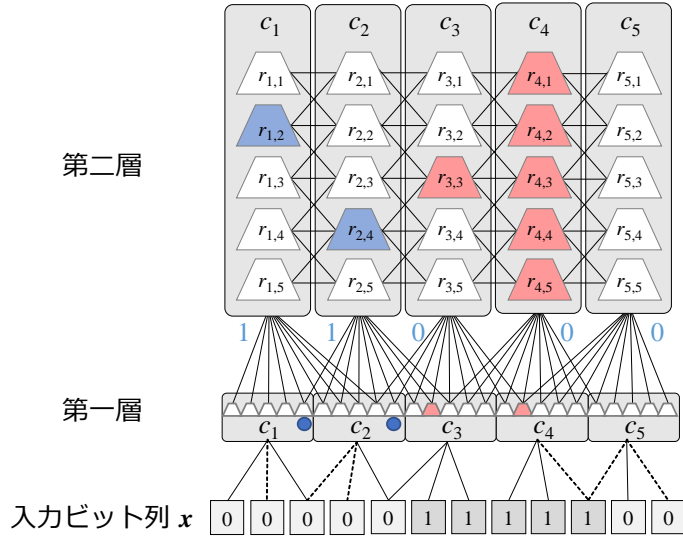


Fig. 2: 提案法：多層 CLA

2.2 アルゴリズム

各時点 t において、以下の手順を実行する。

- 入力変換
- 空間プーリング
- 時間プーリング
- 出力変換

上記の各手順について、以下に述べる。

2.3 入力変換

時刻 t における実数の入力値 $X(t) \in [X^{\min}, X^{\max}]$ をビット列 $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ に変換する。連続する w ビットを 1 にするチャンク変換法を用いる⁶⁾。総ビット長 $n = 12$ 、チャンクビット長 $w = 5$ の例を Fig. 1 に示す。

2.4 空間プーリング

空間プーリングは、入力データビット列 \mathbf{x} を CLA の内部表現にする。具体的には、 \mathbf{x} をカラムの活性パターンに変換する。各カラム c_i について、1 の入力ビットと接続状態にあるシナプス数が多く、活性頻度が低い n_{ac} 本のカラムを活性状態にする。活性状態になったカラムのシナプスの永続値を更新する。1 の入力データビットとのシナプスの永続値を p_c^+ 増加させる。0 の入力データビットとのシナプスの永続値を p_c^- 減少させる。これにより、活性状態のカラム群による入力データビット列の内部表現が強化される。

2.5 時間プーリング

時間プーリングは、セルの活性状態のパターンによって、入力データの文脈を内部表現に変換し、セルの予測状態のパターンによって、次時点 $t+1$ の予測データを内部表現に変換する。具体的には、まず、活性状態の各カラムにおいて、予測状態のセルがある場合はそのセルを活性状態にして、ない場合はカラム内の全てのセルを活性状態にする。これにより、入力データの文脈を内部表現する。予測状態から活性状態になったセルについて、 $t-1$ 時点で活性状態だったセルとのシ

ナプスの永続値を p_r^+ 増加させ、それ以外のシナプスの永続値を p_r^- 減少させる。これにより、前時点 $t-1$ での予測のうち、成功したシナプスが強化される。次に、全てのセルにおいて、時点 t で活性状態のセルと接続状態にあるシナプスが n_{ar} 本以上のセルを全て予測状態にする。これにより、次時点 $t+1$ の予測データを内部表現にする。

2.6 出力変換

セルの予測状態のパターンを入力データビット列と同じ形式にデコードする。本稿では、疎分散表現分類器 (sparse distributed representations classifier, 以下 SDRC)³⁾ を用いる。

3 簡素型 CLA

従来の CLA は、アルゴリズムの複雑さ、パラメータの多さ、実行ごとの動作の不安定さに難点がある。これを改善するため、我々はこれまでに簡素型 CLA を提案した¹¹⁾。簡素型 CLA は、カラムのシナプスに注目し、シナプスの決定論的な配置、初期永続値の固定値化、シナプス配置の集約化が導入されている。本稿では、簡素型 CLA を基礎として、多層 CLA を構築する。

4 提案：多層 CLA

4.1 方法

提案法の概念図を Fig. 2 に示す。まず、第一層では、図の下部に示したように、従来の CLA と同様に入力データビット列を入力として、状態遷移・学習を行い、活性セルを決定する。次に、第二層では、図の上部に示したように、第一層の活性セルを入力として、状態遷移・学習を行い、予測セルを決定する。そして、第一層において、第二層からの予測情報を逆空間プーリング¹²⁾ でフィードバックしながら、予測セルを決定する。第二層によって「第一層で予測状態になる」と予測されたセルを、図においては青い丸で示している。

フィードバックとは、第二層からの予測情報を用いて、第一層の学習を補正するものである。具体的には、第一層で活性状態となったセルが、前時点で第二層から予測されていたセルだった場合、学習の強度を高め

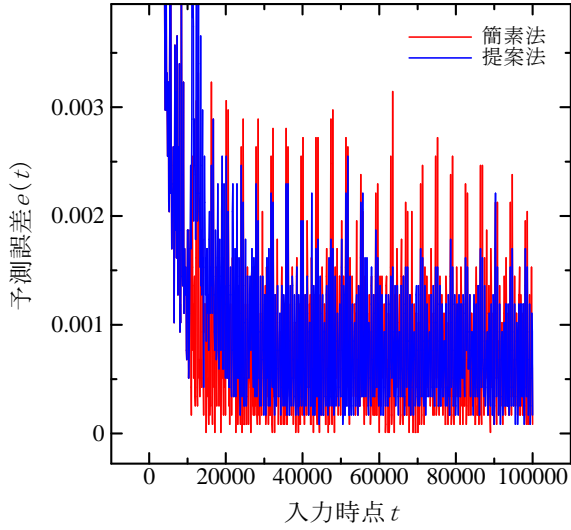


Fig. 3: 正弦波 $X_s(t)$

る。図においては、青い丸が付いたセルが次時点で活性状態になった場合、学習の強度を高める。本稿では、学習の強度を上げる方法として、永続値の増減量 p_r^+ , p_r^- を倍にしている。次に、第一層で予測状態となるセルを決めるとき、第二層から予測されていたセルだった場合、閾値を下げる。図においては、青い丸が付いたセルに対して閾値の緩和を行う。具体的には、予測状態となるために必要な活性状態のセルとの接続シナプス数 n_{ar} を減らし、シナプスが接続状態となる閾値 θ_r を減らす。本稿では、それぞれ半減している。

また、第二層において、従来のカラムのシナプスの初期化では有効なシナプス群を生み出すことは不可能であるため、動的にシナプスを生成する方法を用いる。具体的には、空間プーリングを行った際に、活性状態となるカラムが定められた活性数 n_{ac} よりも少ない場合、活性頻度が低いカラムを選択し、そのカラムに1であるビット全てとのシナプスを持つセグメントを生成する。ちなみに、第一層の学習が不十分な状態でこの方法を用いても有効なシナプス群を生み出すことは難しいため、第二層の学習は10001時点目の入力から行った。

4.2 期待される効果

この方法は、フィードバックによって学習効率を向上させることができるため、学習速度の向上が期待できる。また、単一層では存在しなかったフィードバックが存在することで、第一層の学習に失敗したとしても、第二層からのフィードバックで補正を行うことができるため、そのようなケースにおいて予測精度が向上する効果が期待できる。そして、複雑な入力において、各層でその複雑さを分散できるため、必要な計算領域を削減する効果が期待できる。

5 実験設定

5.1 時系列データ

ベンチマークとなる入力時系列データとして、周期的な時系列データである正弦波 $X_s(t)$ と正弦波の合成波 $X_c(t, n)$ を用いる。また、非周期的な時系列データであるロジスティック写像 $X_l(t, \alpha)$ を用いる。時点 t

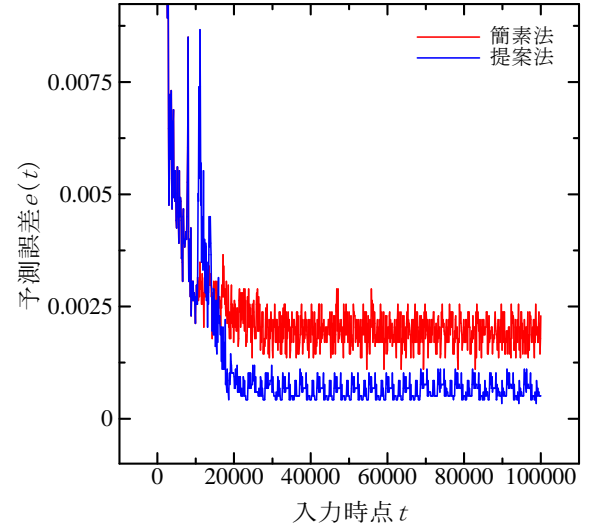


Fig. 4: 合成波 $X_c(t, 2)$

における3種類の入力時系列データを次式で定義する。

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

$$X_c(t, n) = \frac{1}{2} \cdot \sum_{k \in \{1, 3, \dots, 2n-1\}} \frac{1}{k} \cdot \sin\left(\frac{(t-1) \cdot k \cdot \pi}{50}\right) + \frac{1}{2}, \quad (2)$$

$$X_l(t, \alpha) = \begin{cases} 0.4, & \text{if } t = 1, \\ \alpha \cdot X_l(t-1) \cdot \{1 - X_l(t-1)\}, & \text{otherwise.} \end{cases} \quad (3)$$

正弦波の合成波については、 $X_c(t, 2)$, $X_c(t, 3)$, $X_c(t, 4)$, $X_c(t, 5)$ の4種類を用いる。ロジスティック写像については、 $X_l(t, 3.6)$ を用いる。入力時点の範囲は $t \in [1, 10^5]$ に設定した。また、入力値の値域は $[X^{\min}, X^{\max}] = [-0.01, 1.01]$ に設定した。

5.2 評価尺度

予測精度を評価する指標として、次式で求められる100時点ごとの予測誤差 $e(t)$ を用いる。

$$e(t) = \sum_{n=t-99}^t |\bar{X}(n+1) - X(n+1)|, \quad (4)$$

ここで、 $\bar{X}(t+1)$ は次時点 $t+1$ の予測値で、 $X(t+1)$ は次時点の実際の入力値である。また、 $e(t)$ は $t \in \{100, 200, 300, 400, \dots, 10^5\}$ に対して算出する。 $e(t)$ が小さいほど、予測精度が良いといえる。

5.3 パラメータ

入力変換のパラメータとして、入力データビット数 $n = 421$, チャンク長 $w = 21$ に設定した。空間プーリングのパラメータとして、カラム数 $n_c = 2048$, 各カラムのシナプス数 $n_{cy} = 22$, カラムのシナプスの接続・切断を決定する閾値 $\theta_c = 0.1$, 各時点で活性状態となるカラム数 $n_{ac} = 40$, カラムにおけるシナプスの永続値の増加量 $p_c^+ = 0.05$, 減少量 $p_c^- = 0.00025$ に設定した。時間プーリングのパラメータとして、各カラムのセル数 $n_r = 4$, セルのシナプスの接続・切断を決定す

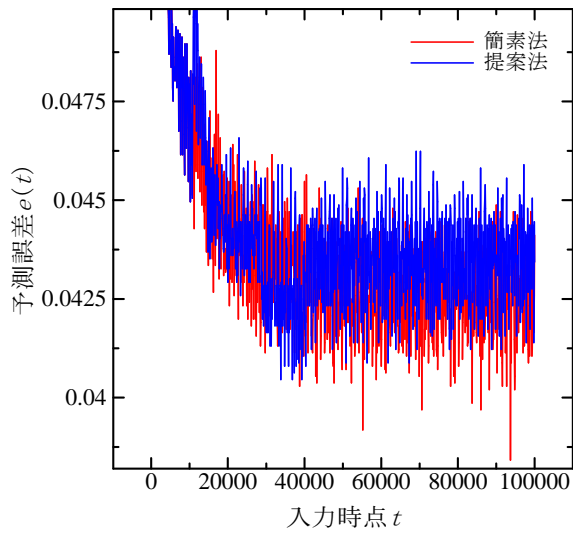


Fig. 5: 合成波 $X_c(t, 3)$

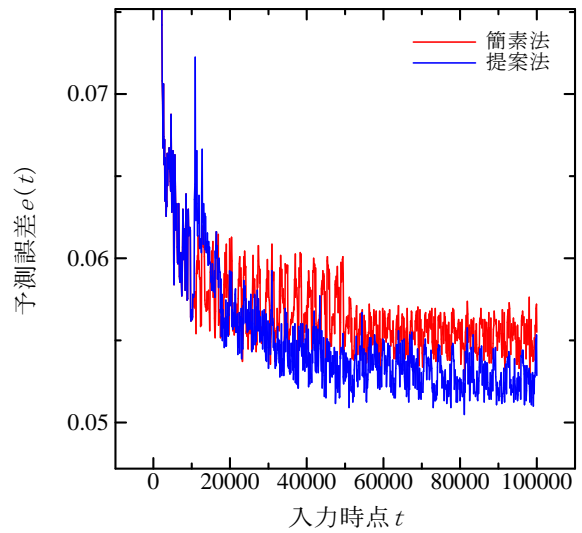


Fig. 6: 合成波 $X_c(t, 4)$

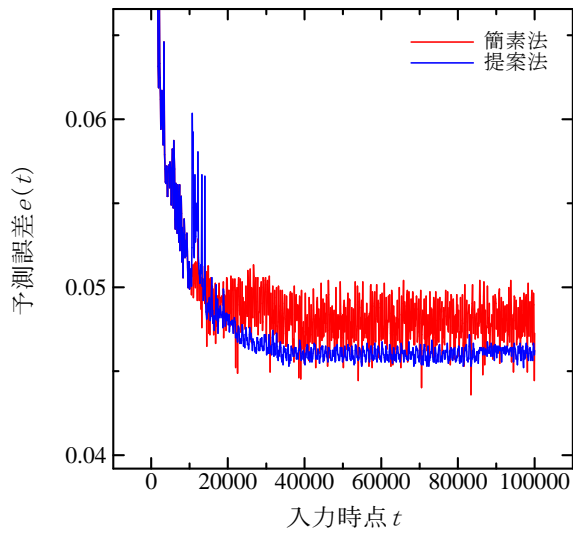


Fig. 7: 合成波 $X_c(t, 5)$

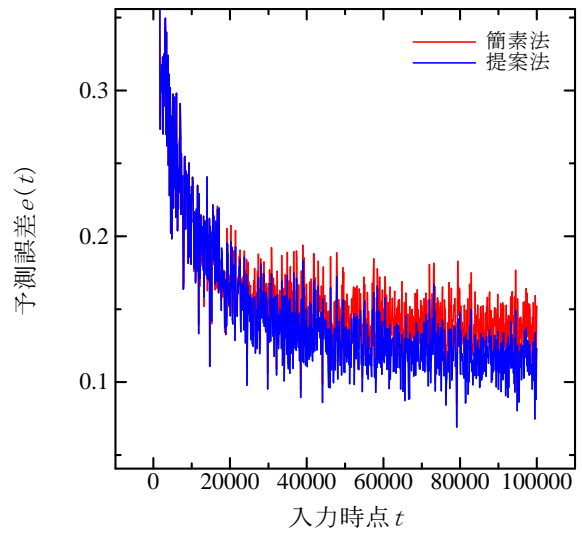


Fig. 8: ロジスティック写像 $X_l(t, 3.6)$

る閾値 $\theta_r = 0.5$, 予測状態となるために必要な接続された活性状態のセル数 $n_{ar} = 15$, セルにおけるシナプスの永続値の増加量 $p_c^+ = 0.1$, 減少量 $p_c^- = 0.1$ に設定した. 多層 CLA においては, 第一層・第二層ともに上記のパラメータを用いる.

6 実験結果と考察

提案する多層 CLA の効果を検証するため, 簡素法と提案法での結果を比較する.

まず, 正弦波 $X_s(t)$ における予測誤差の推移を Fig. 3 に示す. この結果から, 簡素法と提案法の予測誤差は大きく変わらないことがわかる.

次に, 正弦波の合成波 $X_c(t, 2)$ における結果を Fig. 4, $X_c(t, 3)$ における結果を Fig. 5, $X_c(t, 4)$ における結果を Fig. 6, $X_c(t, 5)$ における結果を Fig. 7 にそれぞれ示す. この結果から, Fig. 4, Fig. 6, Fig. 7 より, 提案法の予測精度が $X_c(t, 2)$, $X_c(t, 4)$, $X_c(t, 5)$ で簡素法よりも良いことがわかる. 一方, Fig. 5 より, 提案法の予測精度が $X_c(t, 3)$ で簡素法と比べて良い時点と悪い時点が存在し, 全体としても大きな差が無いことがわかる. そのため, 正弦波の合成波において, 提案法

は簡素法と同程度の予測誤差となるケースが存在するものの, 多くの入力で予測精度を改善するといえる.

そして, ロジスティック写像 $X_l(t, 3.6)$ における結果を Fig. 8 に示す. この結果から, $X_l(t, 3.6)$ において提案法は簡素法よりも良い予測精度を達成する, といえる.

以上の結果より, 提案した多層 CLA が, 従来法よりも予測精度を改善する入力が多く存在することがわかる.

7 まとめ

CLA の時系列データの予測精度を向上することを目的として, 多層 CLA を提案した. 多層 CLA は, 第一層で入力データを受け取り, 第二層で第一層の文脈情報を受け取って学習する. また, その予測結果を第一層へフィードバックし, 第一層における学習を促進する. そのため, 学習速度を向上させることができ, 予測精度を向上させることができる. ベンチマークの時系列データを使用した実験の結果, 多層 CLA が, 単一層の CLA よりも予測誤差を低く安定化させ, 予測精度を改善することを示した.

今後の課題としては、第二層における学習開始のタイミングについて、適切な時点は入力データによって異なると考えられるため、その問題に対応する方法を構築して検証する予定である。

参考文献

- 1) J. Hawkins and S. Blakeslee : On Intelligence. 1/272, Times Books (2004)
- 2) S. Ahmad and J. Hawkins : Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory, Technical report, 1/18 (2015)
- 3) A. M. Zyarah and D. Kudithipudi : Neuromemrisitive Architecture of HTM with On-Device Learning and Neurogenesis, ACM Journal on Emerging Technologies in Computing Systems, Vol. 15, No. 3, Article 24, 24 pages (2019)
- 4) J. Hawkins, A. Subutai, and D. Dubinsky : Hierarchical temporal memory including HTM cortical learning algorithms, Technical report, Numenta, Inc (2010)
- 5) J. Hawkins and A. Subutai : Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex, Frontiers in Neural Circuits, Vol. 10, 1/13 (2016)
- 6) <https://github.com/numenta/nupic>
- 7) J. L. Elman : Finding Structure in Time, Cognitive Science, Vol. 14, Issue 2, 179/211 (1990)
- 8) J. T. Connor, R. D. Martin, and L. E. Atlas : Recurrent Neural Networks and Robust Time Series Prediction, IEEE Transactions on Neural Networks, Vol. 5, No. 2, 240/254 (1994)
- 9) S. Hochreiter and J. Schmidhuber : Long Short-Term Memory, Neural Computation, Vol. 9, No. 8, 1735/1780 (1997)
- 10) Y. Cui, S. Ahmad, and J. Hawkins : Continuous Online Sequence Learning with an Unsupervised Neural Network Model, Neural Computation, Vol. 28, Issue. 11, 2474/2504 (2016)
- 11) 青木 健, 高玉 圭樹, 佐藤 寛之 : 大脳新皮質アルゴリズムの簡素化と予測精度向上に関する検討, SSI 2017, 135/140 (2017)
- 12) T. Aoki, K. Takadama, and H. Sato : Column-based Decoder of Internal Prediction Representation in Cortical Learning Algorithms, SCIS&ISIS (2020)