

大脳新皮質学習におけるカラムとセルの動的構成に関する検討

鈴ヶ嶺 聡哲^{†1} 青木 健^{†1} 高玉 圭樹^{†1} 佐藤 寛之^{†1}

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 [1]。階層時間記憶を具現化したアルゴリズムの 1 つに大脳新皮質アルゴリズム (Cortical Learning Algorithm, 以下 CLA) [1] がある。CLA には、複数のセルを内包するカラム列があり、各カラムは複数の入力データビットとの間にシナプスを持つ。各カラムは、入力データビットとのシナプス接続によって活性化し、カラム列の活性パターンによって入力データを記憶する。また、各セルには活性・通常・予測の状態があり、セル間のシナプスネットワークによってセルの状態を遷移させることで予測する。タクシー乗車数の時系列予測において、CLA は長短期記憶 (LSTM) より高い予測精度を示すことが報告されている [2]。しかし、CLA は、カラムとセル数を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。未知の入力データに対して、適切なカラムとセル数を事前に設定することは困難である。

本研究では、CLA におけるカラムとセルの数の事前設定を不要にすることを目的とし、入力される時系列データに合わせて動的にカラムとセルを生成する方法を提案する。正弦波の合成波、カオスのロジスティック写像を時系列予測する実験において、提案法の効果を検証する。

2. 大脳新皮質学習アルゴリズム (CLA)

CLA の構成を 図 1 に示す。ここでは、5つのカラム、各カラムに5つのセルを用いる例を示す。以下、CLA におけるカラムとセルの役割について述べる。

2.1 カラムの役割

カラムは、入力データビット列を各カラムの活性パターンによって記憶する役目を担う。各入力データビットは、カラムとシナプスで接続される。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する。これにより、入力データビット列がカラムの活性パターンで表現されるようになる。従来の CLA では、カラム数を事前に決定する必要がある。

2.2 セルの役割

セルは、同一の入力データを出現した時系列によって区別する役目と、次の入力を予測する役目を担う。セルは、各カラム内に配置される。活性化されたカラム内から、活性化

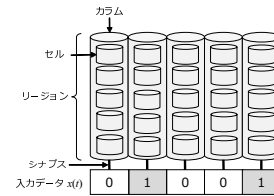


図 1 CLA の構成

せるセルを選択する。活性化されたカラムの中に、予測状態のセルが存在すれば、そのセルを活性化させる。予測状態のセルが存在しない場合、すべてのセルを活性化させる。次に、予測状態にするセルを選択する。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。従来の CLA では、セル数を事前に決定する必要がある。

2.3 問題：適切なカラムとセル数の事前設定

従来の CLA は、カラムとセルの数を事前に設定する必要があるところに問題がある。入力されるデータによって、適切なカラム数とセル数は異なる。特に、時系列情報を保持するセル数が不足すると予測が困難になる。未知の入力データに対して、適切な数のカラムとセル数を事前に設定することは困難である。

3. 提案法：カラムとセルの動的構成

本研究では、CLA におけるカラムとセル数の事前設定を不要にすることを目的とし、カラムとセルを入力データに合わせて動的に追加する方法を提案する。提案法は、入力データの各ビットに 1 つのシナプスを用意し、カラムとセルともに存在しない状態から開始する。

3.1 カラムの追加と活性

入力データビット列における 1 とシナプス接続されたカラムが存在する場合、そのカラムを活性状態にする。存在しない場合、図 2 に示すように、新たにカラムを追加して活性状態にする。このように、各入力データビットにおける 1 の出現にあわせて必要なカラムを追加していく。

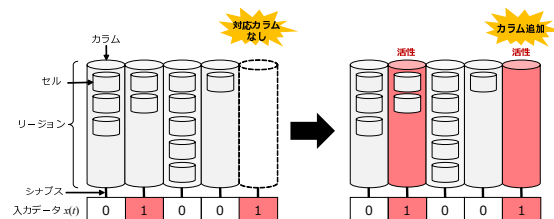


図 2 提案 CLA のカラムの動的生成

^{†1} 現在、電気通信大学
Presently with The University of Electro-Communications

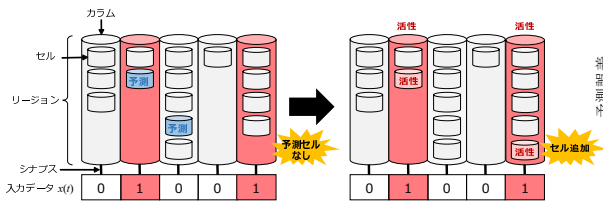


図 3 提案 CLA のセルの動的生成

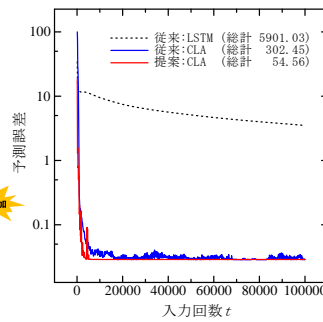


図 4 合成波における結果

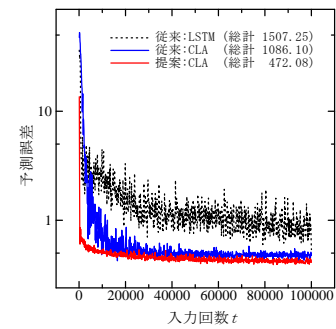


図 5 ロジスティック写像における結果

3.2 セルの追加と活性

活性状態のカラムにおいて、予測状態のセルを活性状態にする。予測状態のセルが存在しない場合は、確率 p で図 3 に示すように、新たなセルを追加し、1 ステップ前に活性状態だったセル群との間にシナプスを形成する。このように、新たに出現した時系列を認識させるセルを徐々に追加していく。一方、セルの過剰な追加を抑制するため、確率 $1-p$ では、セルを追加せず、既存のセルに対して、過去に遡って活性状態にするセルを決定するバックトラック処理を施す。バックトラック処理では、1 ステップ前に活性状態だったカラム内の全てのセルを一時的に仮の活性状態とし、全てのシナプスが仮の活性状態のセルと接続されているセルを列挙する。列挙されたセルが、予測状態のセルが存在しない活性状態のカラムにあるとき、そのセルを活性させる。ただし、バックトラック処理で活性させるセルは、カラムにつき1つとする。

3.3 セルの予測状態化

セルが追加されたカラム内のすべてのセルを一時的に仮の活性状態にする。追加されたばかりのセルは、次のステップを予測するためのシナプスネットワークが未熟なためである。次に、活性状態のセルとのシナプス数が多い順にセルを予測状態にする。この際、予測状態にするセル数は、あらかじめ決めたカラムの活性数と等しくする。また、1つのカラムにつき予測状態にするセルの数は1つとする。

4. 実験結果と考察

本稿では、テスト入力データを用いて従来の LSTM[3]、従来の CLA[1]、提案する CLA の時系列予測の精度を比較する。テスト入力データとして、式 (1) で表される正弦波の合成波、式 (2) で表されるロジスティック写像を用いる。

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

$$x(t) = a \cdot x(t-1) \cdot \{1 - x(t-1)\} \quad (2)$$

入力期間は、 $t \in [1, 10^5]$ とする。式 (2) のロジスティック写像では、 $a = 4.0$ 、 $x(0) = 0.1$ とした。入力データの実数値は、2進数に変換してそれぞれの学習器に入力する。評価尺度として、予測誤差を用いる。真値と予測値の絶対誤

差を入力回数 100 回ごとに合計し、小さいほどよい結果と判断する。試行回数は 30 回とし、予測誤差の平均値と比較する。LSTM[3] では、隠れ層のユニット数を 100 に設定した。損失関数は平均二乗誤差を用い、最適化アルゴリズムは合成波では SGD、ロジスティック写像では Adam を採用した。提案する CLA では、 $p = 0.8$ を用いた。

入力回数 t に対する予測誤差の推移を図 4 と 5 に示す。すべての入力に対して予測誤差を合計した値も併せて図中に記載した。図 4 から、2つの CLA が LSTM より予測誤差の減少が早く、予測誤差の合計も小さいことがわかる。これより、CLA の予測精度の高さが確認された。2つの CLA を比較すると、従来の CLA は、初期の予測誤差が大きいが、提案する CLA は、初期の予測誤差を削減できることがわかる。また、それ以降も、提案する CLA は小さな予測誤差で推移することがわかる。予測誤差の総計は、提案する CLA が最も小さいことがわかった。次に、図 5 においても、提案する CLA は、初期の予測誤差、予測誤差の収束の速さと収束値、予測誤差の総計の観点で最も良好な結果を示すことが確かめられた。このように、提案する CLA は、カラムとセルを動的に構成するだけでなく、予測精度も向上することがわかった。

5. まとめ

本研究では、大脳新皮質学習アルゴリズムにおけるカラムとセル数の事前設定を不要にすることを目的とし、カラムとセルを入力データに合わせて動的に構成する方法を提案した。実験の結果、提案法は、カラムとセルを動的に構成するだけでなく、時系列予測の精度も高められることがわかった。今後の課題として、 p の適切な設定法の検討が挙げられる。

参考文献

- [1] J. Hawkins, A. Subutai, and D. Dubinsky, "Hierarchical temporal memory including HTM cortical learning algorithms," *Technical report*, Numenta, Inc, 2010.
- [2] 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.
- [3] Keras: Deep Learning for humans, <https://github.com/keras-team/keras> (2018/1/29 access)