

—論文題目—

# 接続セグメントに時間軸を導入した 階層的時間記憶モデル

Hierarchical Temporal Memory With  
Time Axis Introduced Connection Segments

指導教授

萩原 将文 教授

学習指導副主任

今井 倫太 教授

慶應義塾大学 理工学部 情報工学科

平成 29 年度

学籍番号 61412912

内藤 慎一郎

# 目次

あらまし	1
第1章 はじめに	2
第2章 Hierarchical Temporal Memory(HTM)	5
2.1 HTM の概要	5
2.2 HTM の構造	5
2.2.1 セルの状態変化	6
2.2.2 パターン表現	7
2.2.3 セル内の構造	7
2.2.4 疎な分散表現	8
2.3 HTM の学習アルゴリズム	8
2.3.1 活性化状態の計算	9
2.3.2 予測状態の計算	9
2.3.3 セグメント集合を用いた接続値の更新	10
2.4 HTM の学習の例	11
2.4.1 学習前	11
2.4.2 学習中	12
2.4.3 学習後	13
2.4.4 並列同時予測	14
2.5 エンコーダと分類器	15
2.6 HTM の問題点	15
第3章 時間軸セグメントを導入した HTM	17
3.1 提案モデルの概要	17

3.2	提案モデルの構造 . . . . .	18
3.2.1	従来型と同じ部分 . . . . .	18
3.2.2	セル内の構造 . . . . .	18
3.3	提案モデルの学習アルゴリズム . . . . .	19
3.3.1	予測状態のセルの計算 . . . . .	19
3.3.2	セグメント集合を用いた接続値の更新 . . . . .	19
<b>第 4 章</b>	<b>評価実験</b>	<b>20</b>
4.1	実験 1 . . . . .	20
4.1.1	実験目的 . . . . .	20
4.1.2	実験概要 . . . . .	20
4.1.3	実験条件 . . . . .	21
4.1.4	実験データ . . . . .	22
4.1.5	学習方法 . . . . .	23
4.1.6	評価指標 . . . . .	24
4.1.7	実験結果 . . . . .	24
4.1.8	実験考察 . . . . .	26
4.2	実験 2 . . . . .	27
4.2.1	実験目的 . . . . .	27
4.2.2	実験概要 . . . . .	27
4.2.3	実験条件 . . . . .	27
4.2.4	実験データ . . . . .	27
4.2.5	学習方法 . . . . .	28
4.2.6	評価指標 . . . . .	28
4.2.7	実験結果 . . . . .	29
4.2.8	実験考察 . . . . .	32
<b>第 5 章</b>	<b>結論</b>	<b>34</b>
	<b>謝辞</b>	<b>35</b>

参考文献	36
付録	38
付録 A 実験 1 における $\sin$ 波の再現例	38
付録 B HTM の実装	44

## あらまし

本論文では大脳皮質の構造と学習アルゴリズムを模した時系列予測モデルである Hierarchical Temporal Memory(HTM) に対して、長期依存関係を考慮できるような改良案を提案する。HTM は脳の神経細胞を表すセルを 2 次元マップ上に配置し、活性化状態になったセルの集合によりデータを表現する。また予測状態になったセルの集合により次の時刻のデータを予測する。各セル間を繋げるシナプス接続をヘブ則に基づいて更新することでデータ同士の関係を保持し、時系列データを学習する。従来のモデルでは時系列データ中の各データに対して一時刻前のデータとの接続のみを学習していたが、提案モデルでは複数時刻前のデータとの接続を学習させることを目的とした。具体的には構造と学習アルゴリズムの両面で従来の HTM を改良したモデルを提案する。構造ではシナプスの集まりであるセグメントに対して時間軸を導入した。学習アルゴリズムではセルの予測状態への遷移において複数時間前のデータとの接続を利用する。実験により提案モデルは時系列データの予測タスクにおいて従来モデルよりも高い精度を記録することを確認した。

# 第 1 章

## はじめに

時系列データの分析や予測は様々な分野で研究が進められてきた。その中でも主に用いられてきたのは統計的手法と機械学習である。統計的手法では古くからある隠れマルコフモデル [8] を時系列データの予測に用いたもの [10] や、自己回帰和分移動平均モデル [3] などが挙げられる。機械学習ではニューラルネットワークを用いた研究が進められている。近年、特に深層学習という多層に重ねたニューラルネットワークが時系列データを扱う様々なタスクで成果をあげている。時系列データを扱うために初めて考えられたニューラルネットワークモデルは RNN[9] である。RNN はニューラルネットワークの中間層を再帰させているため中間層の数が大きくなり、勾配消失問題が発生すると言われている。この問題に対処するために時系列データを扱う様々なニューラルネットワークが研究されている。その中で LSTM[11] や ESN[2] が高い精度を記録している。LSTM は RNN にゲート機構をもたせることで必要な情報のみを伝播させることで勾配消失の低減を目指している。ESN は中間層を再帰させるのではなく中間層をつなぎ替えることで時系列データを学習させた。また LSTM は忘却ゲートの導入 [1] によって長期依存関係の考慮において高い精度を記録している。

しかし人間の脳における大脳皮質の学習は深層学習のように目的関数の誤差を減少させることで最適化を行っているのではないとされている [4]。人間の脳における学習はニューロンが発火していくことによるニューロン間をつなぐシナプスの発生によって行われる。この学習を模したニューラルネットワークモデルとして Hierarchical Temporal Memory(HTM) が提案された。[12]

HTM の構造や学習アルゴリズムは人間の脳の大脳皮質を模したものとなって

いる。HTM の構造は神経細胞を模したセルと名付けられたノードを 2 次元マップ状に配置したものとなっている。またそのセルはそれぞれ非活性化状態、予測状態、活性化状態と遷移するようになっている。学習アルゴリズムはそのセル間の繋がりであるシナプス接続の可否を変化させるというものである。シナプス接続の可否は各シナプスの接続値を増減させることで変化させるが、それはヘブ則によって行われる。これはあるセルの発火の後に他のセルが発火した時にその 2 つのセル間のシナプス接続を強化させるというものである。

HTM はシナプス接続を元にして活性化状態のセルと繋がっているセルが予測状態に遷移し、次の時刻に置いて予測状態のセルが活性化状態に遷移することで時系列データを再現しようとする。HTM はこのシナプス接続を束ねたものとしてセグメントという機構を持つが、これは神経細胞の樹状突起を模したものである。このセグメントは 1 つのセルに対して複数あるためセグメント集合となる。HTM のように神経細胞の働きを模して時系列データの学習を行うモデルをスパイクニューロンモデル [7] という。これは人間の脳の第一視覚野や聴覚野でも行われているとされている。[6]

HTM の学習は深層学習でよく用いられるミニバッチ学習とは異なり、1 つ 1 つのデータの入力に対して学習を更新するオンライン学習となっている。また目的関数に対して最適化を行うのではないため教師なし学習に分類される。

HTM の他の特徴として疎な分散表現が挙げられる。マップ状になったセルの内のわずかなセルのみが発火してデータを表現するため少ない計算量と幅広い表現力を持っている。

HTM は深層学習におけるニューラルネットワークに対して限られたタスクにおいては優位性があるが、ほとんどのタスクにおいて低い精度となっている。また自然言語処理などの複雑なタスクには用いられていない。特に長期依存考慮のための忘却ゲート機構を導入した LSTM[1] のような高度な時系列予測が行えていない。

この原因として HTM には長期依存考慮のための機構を備えていないことが考えられる。特に疎な分散表現による学習が続けられていくに従って発火するセルが減少する。これによって予測情報が維持できなくなり、長期依存関係が

消失する。

そのため本研究では HTM のセグメント集合に対して時間軸を導入し、複数前の時刻におけるデータを表現しているセルとの繋がりも学習可能なモデルを構築する。これによって疎な分散表現を維持しつつも長期依存関係の学習を目指す。人間の脳においてニューロンの発火は完全に同期的ではなく、ある時間ステップに基づいて発火している訳ではない。そのためニューロンの発火時間のずれによってある一定の時間に発火したニューロン全てとのシナプス接続を学習すると考えられる。本研究における HTM の改良はこのニューロンのシナプス接続の時間的に曖昧な部分を模したものになるとも言える。また HTM の学習アルゴリズムを改良した研究というものは現在挙げられていない。これを本研究の大きな貢献とする。

本研究の評価として様々な時系列データの予測タスクにおいて従来の HTM との比較を行った。

以下、第 2 章で HTM の構造と学習アルゴリズムについて詳しく述べ、第 3 章で改良した HTM について、第 4 章で評価実験、第 5 章で結論を述べる。



## 第 2 章

# Hierarchical Temporal Memory(HTM)

## 2.1 HTM の概要

HTM は大脳皮質の構造と学習アルゴリズムを模して作られたニューラルネットワークである。構造はカラムとセルによる 2 次元マップ表現となっており、各セルが状態を偏移させる。学習アルゴリズムはヘブ則となっており各時刻間で活性化状態のセル同士の接続を強める。また活性化状態のセルと強く接続しているセルが予測状態に遷移することで予測を行う。

HTM は全部のセルを用いて 1 つのパターンを表現する。各セルの状態が遷移することで表現するパターンが変化する。

HTM の特徴は時系列データ中の各パターンを再現しつつ学習するためオンライン学習を行っているという点とモデルの訓練とデータの再現を同時に行っているという点がある。これによって時系列データを連続してモデルに流し続け学習を行うという連続オンライン学習が可能となっている。また疎な分散表現を用いることによる並列同時予測が可能な点も挙げられる。

## 2.2 HTM の構造

HTM の全体構造を図 2.1 に示す。HTM は複数のカラムの集合となっており、各カラムは複数のセルの集合からなっている。HTM 全体で 1 つのパターンを表現する。時間が進むごとにパターンが変化していくことで時系列データを表現する。

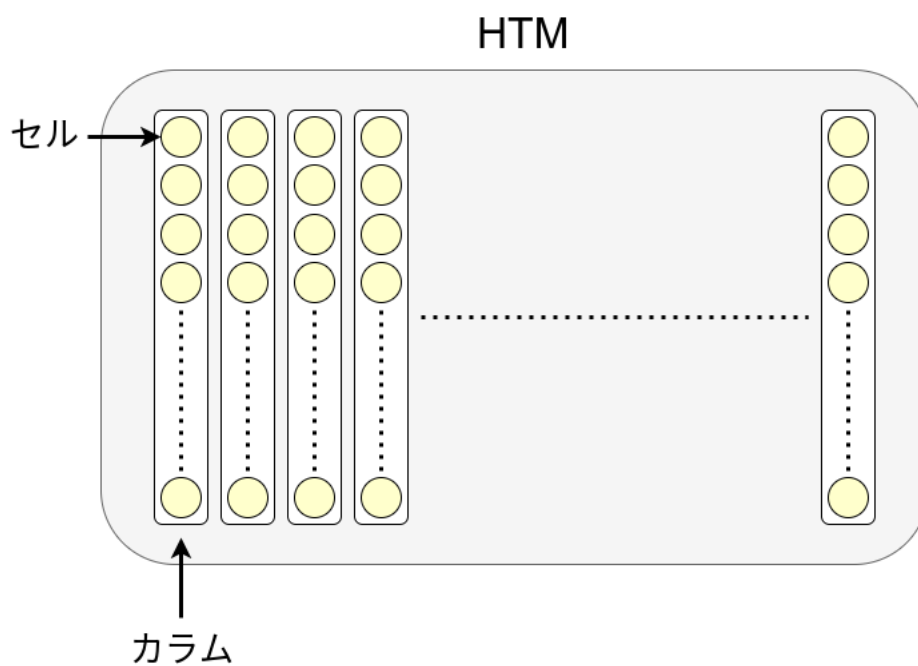


図 2.1 HTM の全体構造

### 2.2.1 セルの状態変化

セルの状態は活性化状態と非活性化状態と予測状態の3つであり、各時刻において常にどれか1つの状態を取る。各時刻ごとにそれぞれの状態を相互に遷移する。

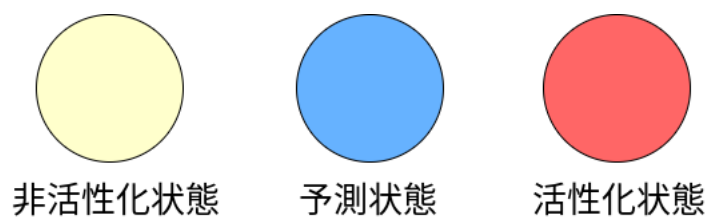


図 2.2 セルの状態

### 2.2.2 パターン表現

HTM はカラムの組み合わせでパターンを表現する。各カラムにおいてカラム中のセルのどれか 1 つでも活性化状態になっている場合にそのカラムがパターン表現を構成するカラムとみなす。

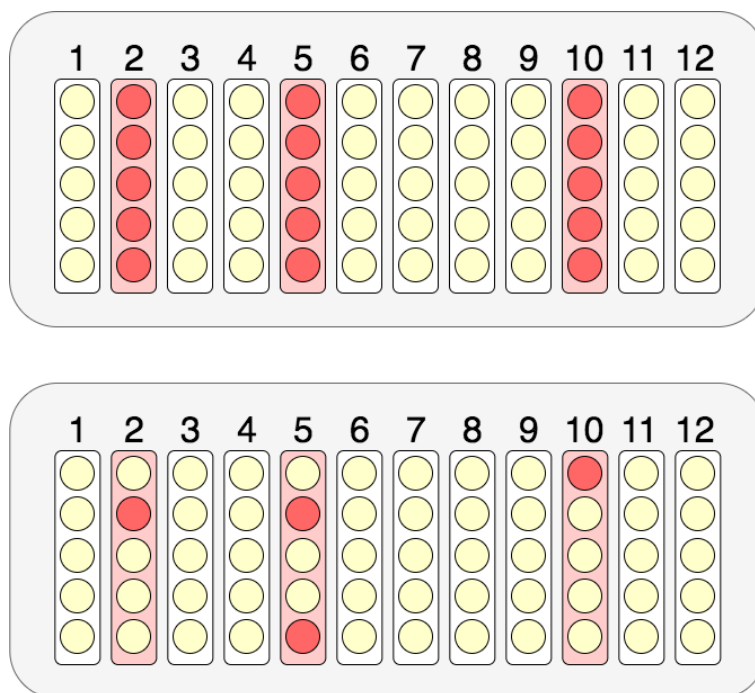


図 2.3 パターンの表現：上下どちらの図も同じカラムの組み合わせ (2, 5, 10) でパターンを表現しているため、両方同じパターンを表す。

### 2.2.3 セル内の構造

HTM はセル内にセグメント構造を持っている。これは大きく分けて、入力セグメントとセグメント集合の 2 つからなる。入力セグメントは入力を受け取るもので、各セルに 1 つある。セグメント集合も各セルに 1 つあり、接続セグメントを複数持っている。この接続セグメントは他のセルとの接続値を保持しており、学習によってこの接続値を増減させしきい値との比較によってセル間の

シナプス接続を判定する。接続セグメントは HTM 中のすべてのセルとの接続値を持つため 2 次元のテンソル値となる。そのためセグメント集合は 3 次元のテンソル値となる。

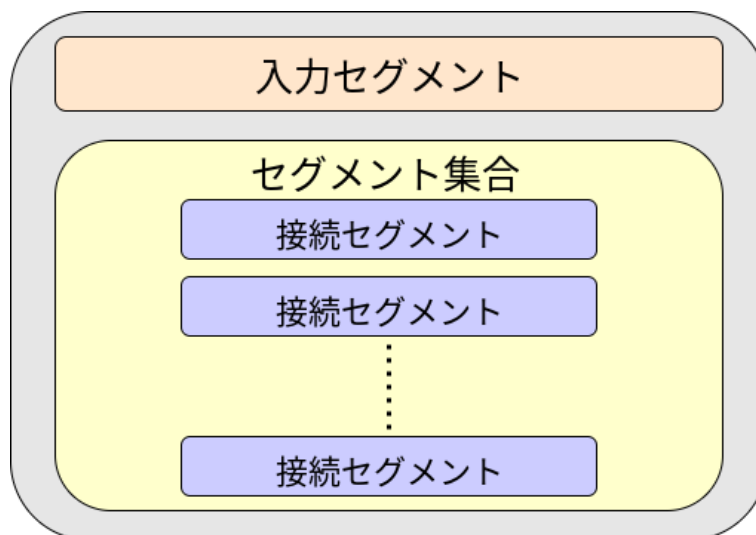


図 2.4 セル内の構造

#### 2.2.4 疎な分散表現

学習が進んでいくと HTM 中の僅かなセルのみが活性化状態に遷移していき、少ないセルでパターンを表現するようになる。したがってあるパターンを表現するのに疎な分散表現を用いていると言える。

### 2.3 HTM の学習アルゴリズム

HTM 中のカラムの数を  $N$  とし、1 カラム中のセルの数を  $M$  とする。このとき HTM 中のセルの総数は  $M \times N$  となる。

時刻  $t$  における活性化状態にあるセルの集合を  $M \times N$  の 2 進行列  $\mathbf{A}^t$  で表し、活性化状態にある  $j$  番目のカラムにある  $i$  番目のセルを  $a_{ij}^t$  で表す。同様に時刻

$t$ における予測状態にあるセルの集合を  $M \times N$  の2進行列  $\Pi^t$  で表し、予測状態にある  $j$  番目のカラムにある  $i$  番目のセルを  $\pi_{ij}^t$  で表す。

各セルにあるセグメント集合の中で  $d$  番目のセグメントの  $j$  番目のカラムにある  $i$  番目のセルを  $\mathbf{D}_{ij}^d$  で表す。またセグメント集合中のセグメントはそれぞれ各セルから他のセルへの接続値を保持しているため、セグメントの接続値がしきい値を超えたセルのみを記録した2進行列  $\tilde{\mathbf{D}}_{ij}^d$  で接続の可否のみを表す。

入力セグメントに入力を受けたセルが存在するカラムの集合を勝者カラムとし、 $\mathbf{W}^t$  と表す。

### 2.3.1 活性化状態の計算

活性化状態の計算は式 (2.1) で表される。

$$a_{ij}^t = \begin{cases} 1 & \text{if } j \in \mathbf{W}^t \text{ and } \pi_{ij}^{t-1} = 1 \\ 1 & \text{if } j \in \mathbf{W}^t \text{ and } \sum_i \pi_{ij}^{t-1} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.1)$$

式 (2.1) は2つの場合においてセルが活性化状態に遷移することを表している。1つ目はセルが勝者カラム中のカラムに含まれており、そのセルが予測状態に遷移している場合である。2つ目はセルが勝者カラム中のカラムに含まれており、そのセルを含むカラム中に予測状態に遷移しているセルが1つも存在しない場合である。図 2.3 におけるパターンの表現において、上の図は2つ目の条件のみによって活性化状態に遷移した場合で、下の図は1つ目の条件のみによって活性化状態に遷移した場合である。実際の学習では2つの条件による活性化状態が混在した状態でパターンが表現される。

### 2.3.2 予測状態の計算

予測状態の計算は式 (2.2) で表される。

$$\pi_{ij}^t = \begin{cases} 1 & \text{if } \exists_d \|\tilde{\mathbf{D}}_{ij}^d \circ \mathbf{A}^t\|_1 > \theta \\ 0 & \text{otherwise} \end{cases} \quad (2.2)$$

式 (2.2) はセルが予測状態に遷移する場合を表している。 $\theta$  は接続しているシナプス接続の数におけるしきい値となっている。このしきい値を超えている時にセルとセルが接続しているとする。セグメント集合中のある接続セグメントに関して活性化状態にあるセルとその接続セグメントによって接続しているセルが予測状態に遷移する。

### 2.3.3 セグメント集合を用いた接続値の更新

セグメント集合が持つ接続値を更新する場合として以下の3つが挙げられる。

1. セルが予測状態になり、その後活性化状態に遷移した場合

$$j \in \mathbf{W}^t \quad \text{and} \quad \exists_i (\pi_{ij}^{t-1}) > 0 \quad (2.3)$$

2. 勝者カラム中にあるどのセルも予測状態になっていなかった場合

$$j \in \mathbf{W}^t \quad \text{and} \quad \forall_i (\pi_{ij}^{t-1}) = 0 \quad (2.4)$$

3. セルが予測状態になっていたが、その後活性化状態に遷移しなかった場合

$$j \notin \mathbf{W}^t \quad \text{and} \quad \exists_i (\pi_{ij}^{t-1}) > 0 \quad (2.5)$$

1 と 2 の場合はシナプス接続を強化する。つまりセグメント集合中にある接続値を増加させる。1 の場合はセグメント集合の中ですでに接続値がしきい値を超えている接続セグメントに関して、2 の場合はセグメント集合のなかで一番大きい接続値を持つ接続セグメントに関して下の更新式を適用する。

$$\Delta \mathbf{D}_{ij}^d = p^+ \dot{\mathbf{D}}_{ij}^d \circ \mathbf{A}^{t-1} - p^- \dot{\mathbf{D}}_{ij}^d \circ (\mathbf{1} - \mathbf{A}^{t-1}) \quad (2.6)$$

$p^+$  と  $p^-$  は学習率となっており、 $p^+$  は  $p^-$  より大きい値となっている。

$\dot{\mathbf{D}}_{ij}^d$  は  $\mathbf{D}_{ij}^d$  の正の値をもつ位置のみをとりだした2進行列である。

○は要素積を表し、行列の要素同士の積をとる演算である。

$$\dot{\mathbf{D}}_{ij}^d = \begin{cases} 1 & \text{if } \mathbf{D}_{ij}^d > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

この更新式は不活性シナプスの接続値を減少させ、活性シナプスの接続値を増加させることによってシナプス接続を強化している。

3の場合はセグメント集合の中でしきい値を超えている接続値を減少させる。これはシナプス接続の減衰を再現している。

$$\Delta \mathbf{D}_{ij}^d = p^{--} \dot{\mathbf{D}}_{ij}^d \text{ where } a_{ij}^t = 0 \text{ and } \|\tilde{\mathbf{D}}_{ij}^d \circ \mathbf{A}^{t-1}\|_1 > \theta, \text{ where } p^{--} \ll p^- \quad (2.8)$$

## 2.4 HTM の学習の例

### 2.4.1 学習前

例として図 2.5 を考える。

学習前の HTM は上の図 2.5 のようになっており、入力に対して予測状態になるセルが存在せずに勝者カラムのセルがすべて活性化状態に遷移していく。

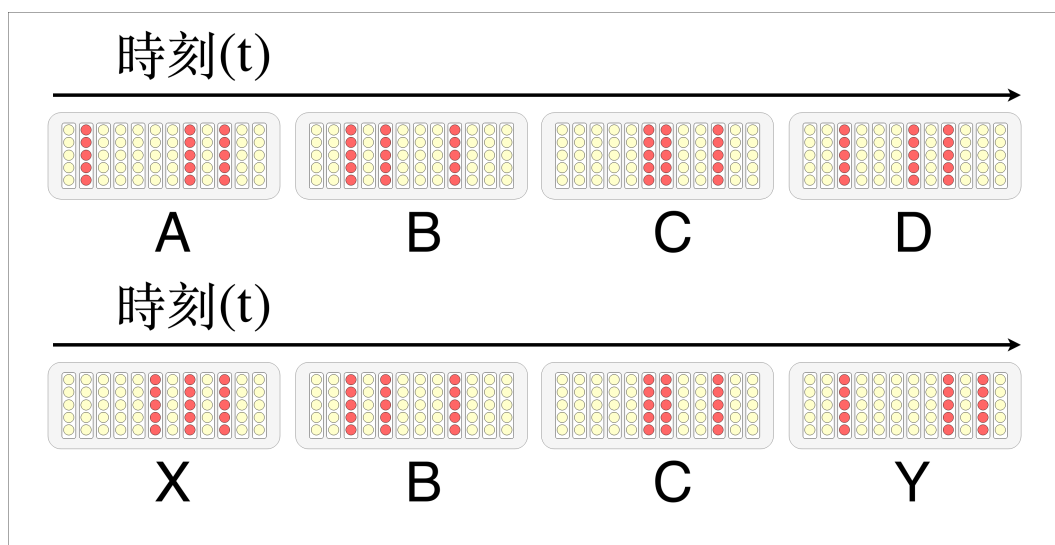


図 2.5 学習前の HTM : (A, B, C, D) という系列と (X, B, C, Y) という系列を入力した場合

### 2.4.2 学習中

例として図 2.6 を考える。

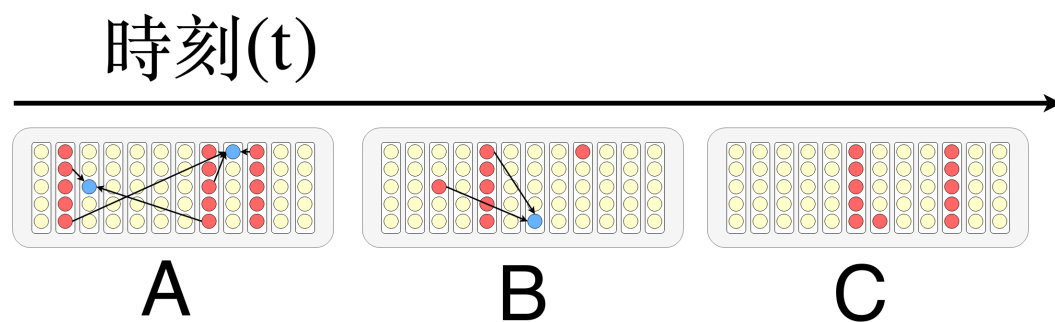


図 2.6 学習中の HTM

学習中の HTM は上の図 2.6 のようになっており、1 つ前の時刻の活性化状態のセルとのシナプス結合によってセルが予測状態に遷移することによって次の時刻のパターンが予測される。また予測状態に遷移するセルが増加するほどパターンの表現が疎になっていく。



### 2.4.3 学習後

例として図 2.7 を考える。

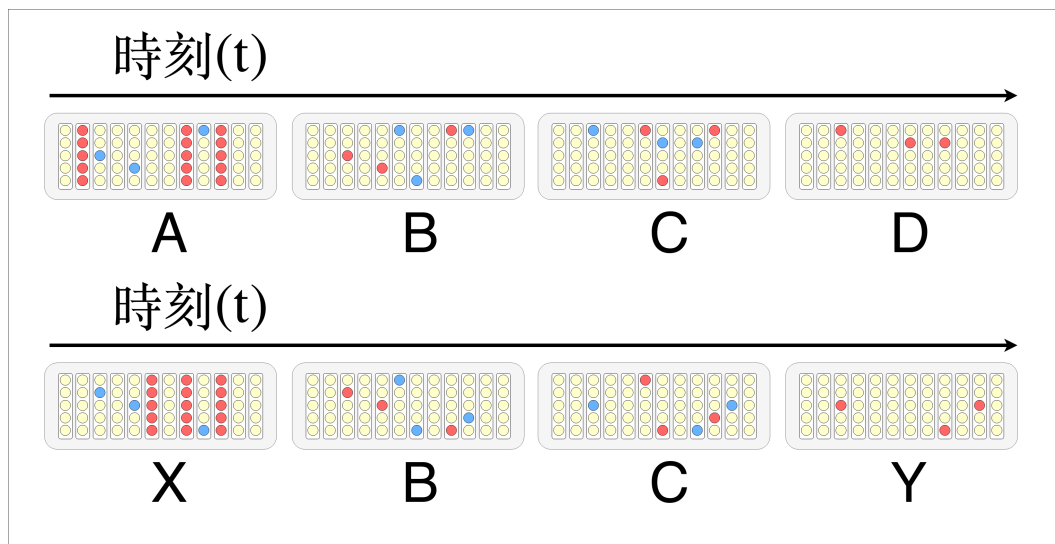


図 2.7 学習後の HTM : (A, B, C, D) という系列と (X, B, C, Y) という系列を入力した場合

学習が完了すると最初の入力以外は疎な分散表現によってパターンを表現するようになり、系列中のすべてのパターンが予測されるようになる。

#### 2.4.4 並列同時予測

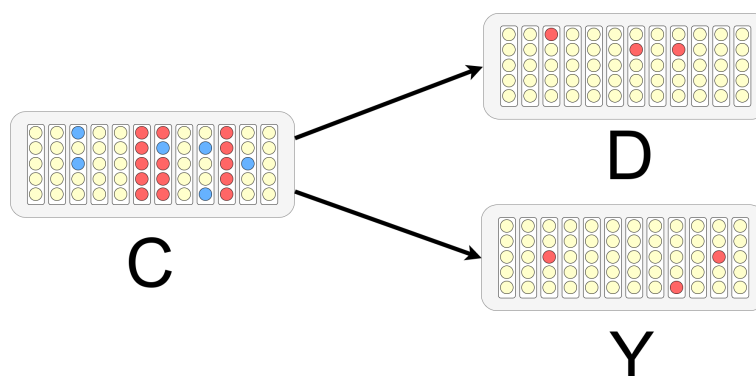


図 2.8 学習後の HTM : (A, B, C, D) という系列と (X, B, C, Y) という系列を学習後に (C) という系列を入力した場合

HTM の特徴として並列同時予測が可能であるということが挙げられる。例として (A, B, C, D) と (X, B, C, Y) という 2 つの系列を学習後に前の文脈がない状態で (C) という系列を与えたものを考える。このとき C は最初の入力であるため勝者カラムのすべてのセルが活性化状態に遷移する。これによって D と Y 両方の予測を導くセルが予測状態に遷移するため、同時に複数の予想が可能になる。

## 2.5 エンコーダと分類器

HTM ではパターンの表現はカラムの組み合わせによって表される。そのためエンコーダによって入力値をカラムの組み合わせに変換して HTM に入力し、分類器によって HTM の予測状態のセルから予測されているパターンへ分類することが必要となる。

エンコーダは入力値に対してランダムに一定数のカラムを選び出して割り当てている。分類器は最尤予測を用いて分類する。HTM の予測状態のセルを以下の式によって各カラムに関する予測点数  $\mathbf{P}_j^t$  に変換する。

$$\mathbf{P}_j^t = \sum_{i=0}^M \frac{\pi_{ij}^t}{M} \quad (2.9)$$

$\mathbf{P}_j^t$  は長さ  $N$  の 1 次行列となる。

ここで各入力値に対応しているカラムの組み合わせを one-hot 行列に変換し、 $\mathbf{P}_j^t$  との要素積をとる。この値をそれぞれの入力に対する予測点数とし、softmax 関数に入力することで各入力値に対する予測値を確率分布で出力する。

## 2.6 HTM の問題点

HTM の問題点は長期依存考慮に関しての性能が低いことがあげられる。本論文では長期依存考慮の性能を測るために活性化状態の計算に予測状態のセルのみを使用して予測を行った。そのため活性化状態の計算は以下のようにした。

$$a_{ij}^t = \begin{cases} 1 & \text{if } j \in \mathbf{W}^t \text{ and } \sum_i \pi_{ij}^{t-1} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (t = 0) \quad (2.10)$$

$$a_{ij}^t = \pi_{ij}^{t-1} \quad (t \neq 0) \quad (2.11)$$

$t \neq 0$  のときに式 (2.11) を用いたことによってカラムの入力を始めの 1 つのパターン以外は入れずに、予測状態のみを用いてセルの遷移を行うことになる。

この際に長期依存考慮の性能が低くなる大きな理由として以下の2つの問題点がある。

1. 疎な分散表現を用いたために発火するセルが徐々に少なくなり消失する。
2. 学習が大きく進んだパターンにおいて表現が疎になった時に次のパターンに繋がっていたセルが消失するために学習状態が損失する。

1の問題について詳細に述べる。予測状態のセルの計算によって予測状態にあるセルは疎な分散表現となる。学習の際は、勝者カラムの中のセルが1つも活性化状態にない時に、すべてのセルが活性化状態になることによって予測状態が疎になりすぎることを防いでいた。しかし長期依存考慮に関する予測タスクでは、予測状態のセルのみを用いて活性化状態を計算する。そのことによって予測状態になるセルが疎になりすぎてしまい、予測を進めるに従って予測状態になるセルが消失するという問題が発生した。

2の問題について詳細に述べる。これは活性化状態の計算によるものである。学習が進んでいないときは勝者カラム中の大部分のカラムにおいて全てのセルが活性化する。それに対して、学習が進むと勝者カラム中の多くのカラムにおいてカラム中の僅かなセルのみが活性化することになる。これによってその僅かなセル以外とつながっていたセルが予測状態に遷移しなくなるために学習状態が損失するという問題が発生する。

これらの問題に対処するため、本論文では時間軸セグメントを導入した HTM を提案する。

## 第 3 章

# 時間軸セグメントを導入した HTM

### 3.1 提案モデルの概要

従来の HTM は 1 つ前の時刻におけるパターンを表現するセルのみとの接続を持つように学習していたが、前章で述べたように長期依存考慮に関して性能が低いという問題があった。そこで、その問題を解消するために接続セグメントに時間軸を導入することによって、複数前の時刻におけるパターンを表現するセルとの接続も持つように学習させた。

この改良の重要な点は新たに接続をもつようになった時刻の範囲においてのみに長期依存関係を保持できるようになったということではない。HTM の学習において予測状態に遷移するセルに関して適切な密度を保持し続けることは重要な要素である。改良した HTM は複数前の時刻にわたって接続を学習することによって、適切な密度が保持されセルの状態遷移が安定して行われる。それによって新たに接続を持つようになった時刻の範囲をはるかに超える長期依存考慮が可能になった。

前章で挙げた以下の 2 つの問題点に関して、この改良の効果を具体的に述べる。

1. 疎な分散表現を用いたために発火するセルが徐々に少なくなり消失する。
2. 学習が大きく進んだパターンにおいて表現が疎になった時に次のパターンに繋がっていたセルが消失するために学習状態が損失する。

まず 1 の問題に関しては複数前の時刻における活性化状態とそれらのセルとの接続セグメントを用いて予測状態のセルの計算を行い、それによるすべての

予測状態のセルを予測に用いるためにセルの減少が抑制されている。

次に 2 の問題に関しては学習が大きく進んだパターンにおいて表現が疎になった場合でも複数の時刻にわたる接続があることによって学習状態の損失が抑制されている。これは次のパターンの予測に用いる学習が複数の時刻間に分散されているためである。

以上の 2 つの問題点の抑制によって長期依存考慮に関する性能が向上した。

## 3.2 提案モデルの構造

### 3.2.1 従来型と同じ部分

HTM 全体の構造やセルの状態遷移、パターンの表現は従来の HTM を用いた。

### 3.2.2 セル内の構造

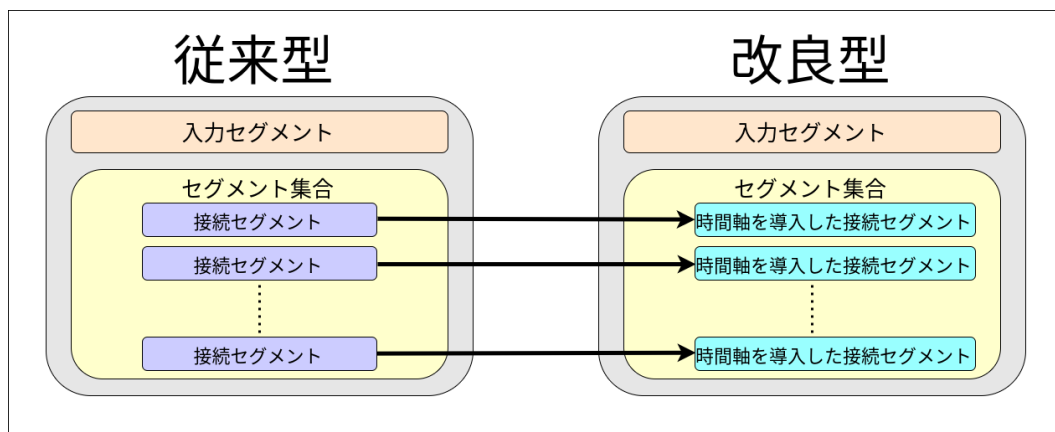


図 3.1 提案モデルにおけるセル内の構造

提案モデルのセル内の構造は図 3.1 のようになっている。入力セグメントは従来型の HTM と同様になっているが、セグメント集合の構造は異なっている。セグメント集合内の接続セグメントに関して時間軸を導入した。これによって

接続セグメントは HTM 中のすべてのセルとの接続値と時間軸の 3 次元テンソル値によって表される。そのためセグメント集合は 4 次元のテンソル値となる。

### 3.3 提案モデルの学習アルゴリズム

提案モデルは従来の HTM を拡張しているため学習アルゴリズムにおいても拡張する必要がある。時間軸を導入したことによって予測状態のセル計算とセグメント集合を用いた接続値の更新において変更を加えた。

#### 3.3.0.1 従来型と同じ部分

活性化状態のセルの計算に関しては従来の HTM と同様である。

#### 3.3.1 予測状態のセルの計算

時間軸セグメントの長さを  $\tau_c$  とし、セグメント集合における接続値を  $\mathbf{D}_{ij}^d(\tau)$  とする。また接続値がしきい値を超えた値のみを取り出し、接続の可否のみを表した 2 進行列を  $\tilde{\mathbf{D}}_{ij}^d(\tau)$  とする。

$$\pi_{ij}^t = \begin{cases} 1 & \text{if } \exists \tau \exists d \|\tilde{\mathbf{D}}_{ij}^d(\tau) \circ \mathbf{A}^t\|_1 > \theta \quad (t - \tau_c < \tau < t - 1) \\ 0 & \text{otherwise} \end{cases} \quad (3.1)$$

#### 3.3.2 セグメント集合を用いた接続値の更新

セグメントを更新する場合分けは従来型と同様となっている。またセグメントの更新式も従来型と同様となっており式 (2.6) と式 (2.7) となっているが、セグメントの更新を複数の時間 ( $t - \tau_c < t - 1$ ) にわたって適用するように変更している。

## 第 4 章

### 評価実験

#### 4.1 実験 1

##### 4.1.1 実験目的

接続セグメントに時間軸を導入した HTM における適切なセグメント集合のサイズの検証

##### 4.1.2 実験概要

改良型の HTM は従来型に比べてセグメント集合の次元が時間軸によって 1 次元拡張されているが、セグメント集合を表すテンソルの大きさは一致するように調整している。HTM におけるセグメント集合の次元は従来型で 3 次元、改良型で 4 次元となっており、これが各セルごとに存在しているが、実装ではすべてのセルのセグメント集合をまとめて定義している。これによって HTM 全体でのセグメント集合は従来型で 5 次元、改良型で 6 次元となっている。本実験ではセグメント集合のサイズをテンソルの合計の大きさを変えずに様々な条件で検定する。

セグメント集合のサイズは以下の 4 つのパラメータによって決まる。

- カラム数
- セル数



- セグメント数
- 時間軸長

カラム数とセル数は接続元のセルと接続先のセルの2つのセルの位置を示すために2度用いられる。セグメント集合のテンソルの合計の大きさ  $S$  は、カラム数を  $N$ 、セル数を  $M$ 、セグメント数を  $d$ 、時間軸長を  $\tau_c$  とすると以下の式で表される。

$$S = N^2 * M^2 * d * \tau_c \quad (4.1)$$

実験に用いた従来型 HTM のパラメータの値は以下の通りである。

表 4.1 従来型 HTM のパラメータの値

パラメータ	値
カラム数 (N)	512
セル数 (M)	16
セグメント数 (d)	32
時間軸長 ( $\tau_c$ )	1

従来型 HTM は接続セグメントに時間軸を導入していないため、時間軸長は1となっている。表 4.1 より、従来型 HTM のセグメント集合のテンソルの合計の大きさ  $S$  は  $S = 512^2 * 16^2 * 32 * 1 = 2^{31}$  となる。

改良型 HTM のセグメント集合のテンソルの合計の大きさは従来型 HTM と同じ  $2^{31}$  と統一した。

### 4.1.3 実験条件

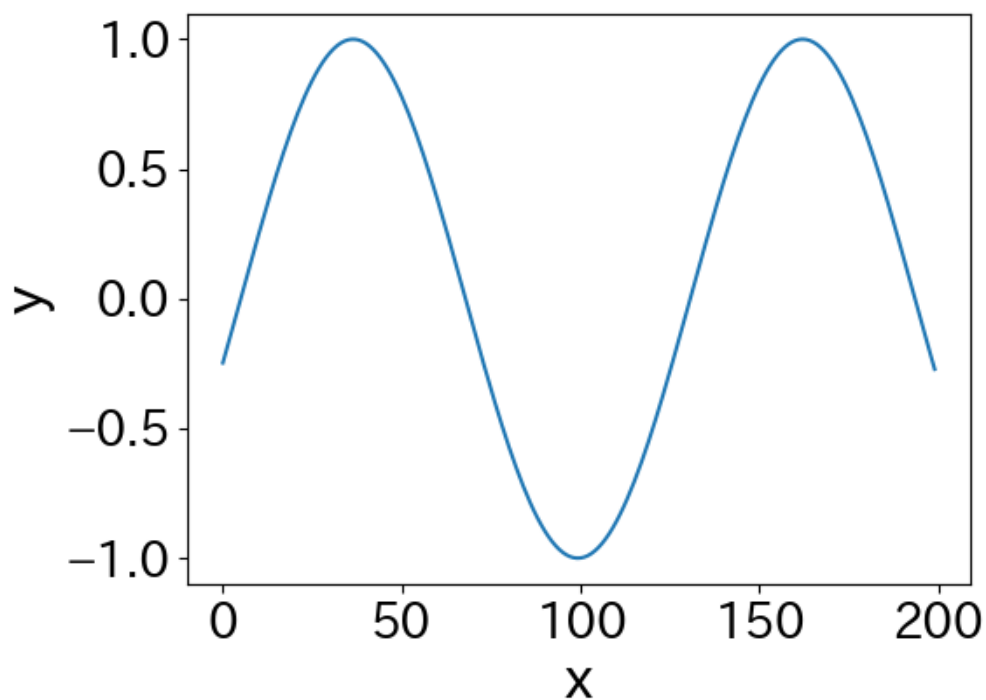
改良型 HTM における適切なセグメント集合のサイズ検定を行うため、以下の表の5通りで実験を行った。

表 4.2 各通りにおける改良型 HTM のパラメータの値

		各条件における値				
		1	2	3	4	5
マ ー ク ス ペ ー ス	カラム数 (N)	512	512	512	512	256
	セル数 (M)	16	16	8	8	16
	セグメント数 (d)	4	8	16	8	8
	時間軸長 ( $\tau_c$ )	8	4	8	16	16

#### 4.1.4 実験データ

本実験における学習データとして周期が  $40\pi$  でデータサンプリング数が 200 の  $\sin$  波を用いた。改良型 HTM における学習にはこの  $\sin$  波の位相をランダムにずらしたものを 5 つ用いた。また最終的な予測精度の測定には学習に用いたものと同様のものを用いた。例として図 4.1 に示す。

図 4.1 実験に用いた  $\sin$  波

#### 4.1.5 学習方法

学習は以下のフローチャートのように行った。

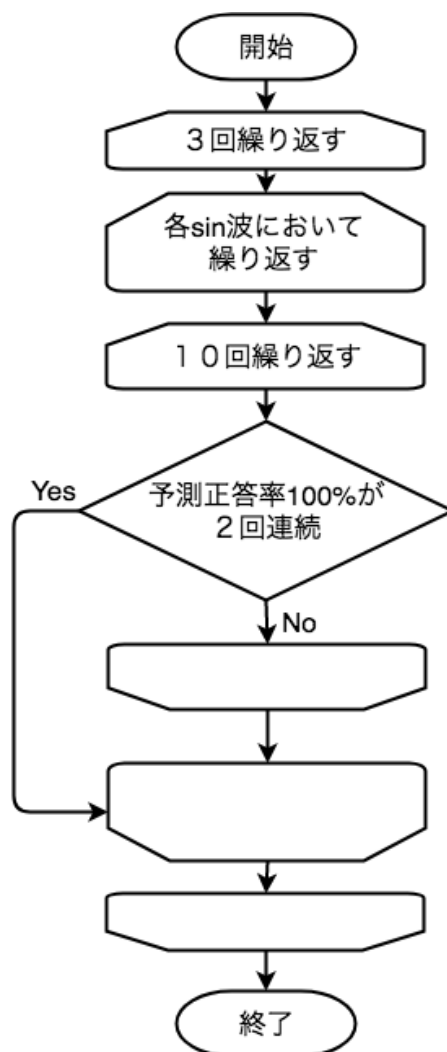


図 4.2 学習のフローチャート

学習方法は HTM に複数のデータを学習をさせる際に 1 つのデータにおけるセル間の接続を確立するために考案した。まず 1 つの *sin* を 10 回繰り返して入力することでその *sin* に対しての HTM の学習を安定させる。また予測正答率が 100% に 2 回連続で到達した時点で学習が完了したとみなし、途中で繰り返しを抜ける。この学習をすべてのデータに対して順に行う。そしてすべての

データに対して学習を行った後、また始めのデータに戻り学習を行う。これを3回繰り返す。途中で繰り返しを抜けなかった場合の学習全体でのエポック数は150回となる。

この学習方法は人間における暗記学習を模している。複数の物事を覚えるときに1つの物事を複数回反芻することで定着させようとする。また全体を一度暗記したら復習を繰り返すことで忘却しにくくする。

#### 4.1.6 評価指標

本実験の評価指標はサンプルデータ数200の $\sin$ 波において、最初の1個のデータを入力したあとに、残りの199個のデータを予測した際の正答率である。

#### 4.1.7 実験結果

本実験における各エポックでのそれぞれの条件での予測正答率を以下の図4.2,4.3に示す。

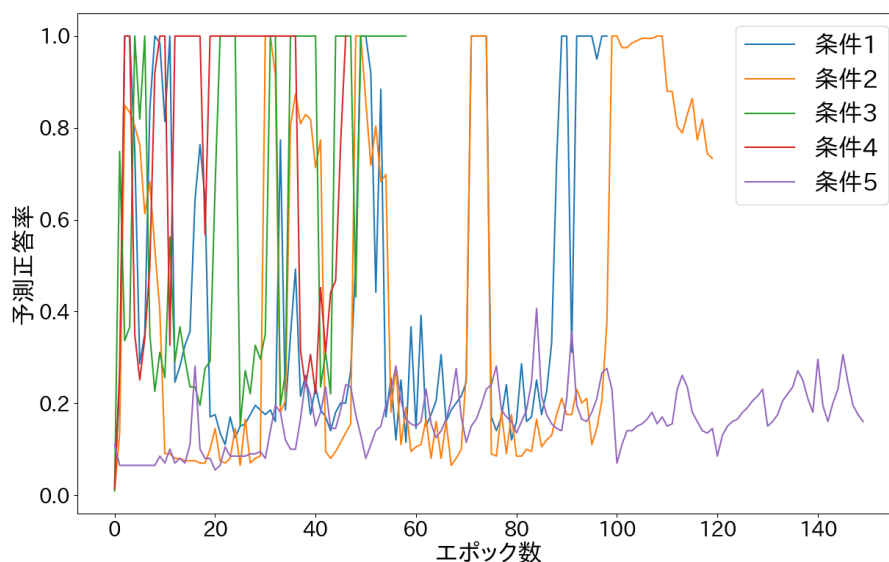


図 4.3 実験1の結果1

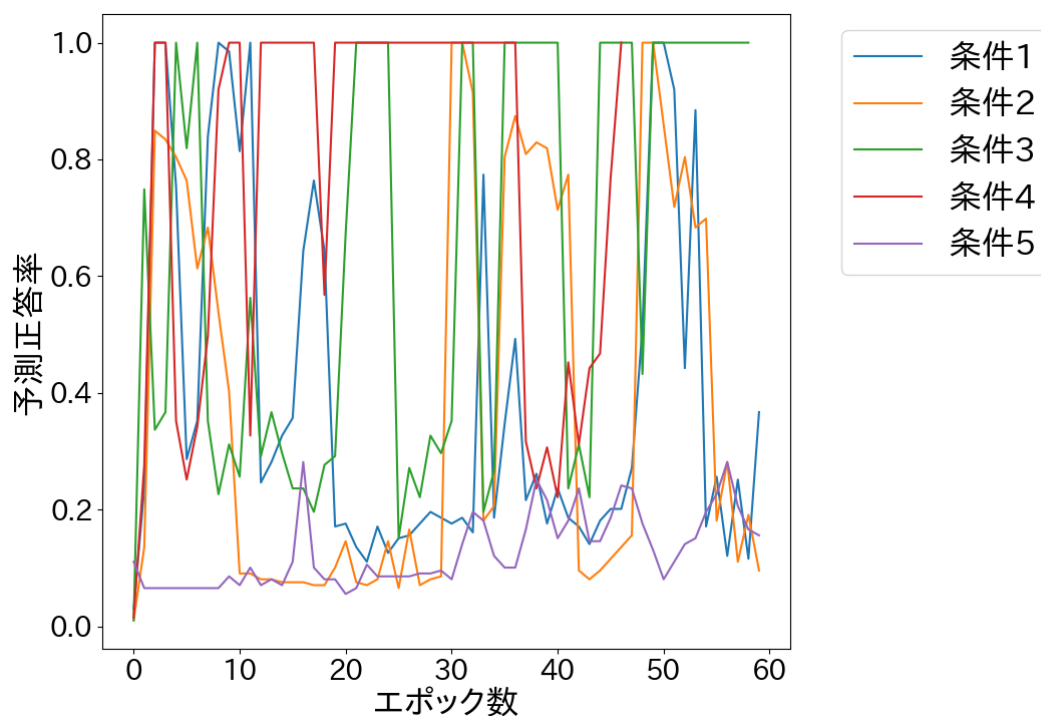


図 4.4 実験 1 の結果 2

図 4.2 は全エポック数における予測正答率を示していて、図 4.3 は 60 エポックまでの予測正答率を示している。HTM の学習における予測正答率は学習の途中においては大きく変化し、最終的には学習によってセル間の接続が確立され、予測正答率が 100% に近づいていく。条件 3 と条件 4 は 60 エポックまでで学習が完了しており、条件 1 と条件 2 は 150 エポック周辺で学習が完了している。また条件 5 は最後まで学習が完了しなかった。

次に学習完了後に学習に用いた 5 つの  $\sin$  波の予測正答率をしたの表 4.3 に示す。

表 4.3 より条件 3 において最も高い予測精度が記録された。また条件 5 で最も低い予測精度となった。

表 4.3 各通りにおける最終予測正答率						
		各条件における予測正答率 (%)				
		1	2	3	4	5
各 <i>sin</i> 波	1	9.0	5.5	100	38.7	0.0
	2	100	74.9	100	16.1	13.6
	3	100	13.6	100	21.6	7.5
	4	100	31.2	100	15.6	8.0
	5	100	73.4	100	100	16.1
平均		81.8	39.7	100	38.4	9.0

4.1.8 実験考察

本実験には以下の 3 つの傾向が現れた。

- 1. カラム数の減少は性能の低下に大きく影響する。
- 2. セル数やセグメント数の減少は性能の低下に影響する。
- 3. 時間軸長の増加は性能の向上に影響する。

1 つ目に関しては、カラム数がパターンの表現に直接影響することによるものだと考えられる。2.2.2 で述べたように HTM におけるパターンの表現はカラムの組み合わせによるものである。カラム数が少なくなると各パターンを表現するカラムに重複するものが多くなる。それによってパターンの誤予測が多くなり、HTM の性能が低下する。

2 つ目に関しては、セル数やセグメント数は学習の柔軟性に関わる部分であるため、少しの減少では学習の精度に影響することはないのだと考えられる。しかし条件 4 のようにセル数とセグメント数の両方を大きく減少させた際に複数のデータを学習すると、セル間の接続の繋がりに偏りが生まれるため、最終的な予測精度が減少すると考えられる。

3 つ目は本研究で導入したものであるが、性能の向上に貢献することがわかった。時間軸長が長いほど性能が高いため、セル数とセグメント数の減少による

性能低下が現れない程度にセル数とセグメント数の分の要素数を時間軸長に割り当てたモデルが最も良いと考えられる。

これらの3つの傾向から以降の実験では条件3のモデルを用いる。

## 4.2 実験2

### 4.2.1 実験目的

改良型 HTM における長期依存考慮の性能検証

### 4.2.2 実験概要

合成関数を学習させた提案モデルを従来型の HTM や機械学習による手法と比較し、長期依存考慮における性能を検証した。

### 4.2.3 実験条件

- 提案モデル
- 従来型 HTM
- RNN
- LSTM
- GRU [5]

### 4.2.4 実験データ

本実験における学習データとして以下のような合成関数を用いた。

$$\sin(x) + \frac{1}{2} \cos(2x) + \frac{1}{3} \sin(4x) + \frac{1}{4} \sin\left(\frac{x}{4}\right) \quad (4.2)$$

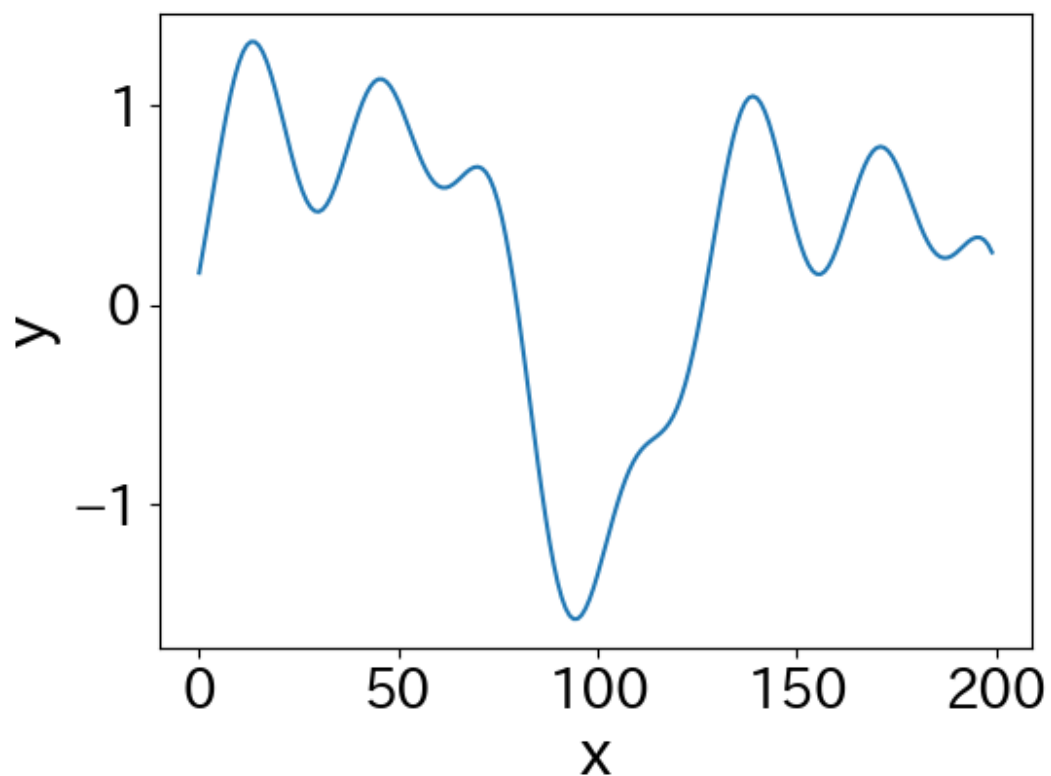


図 4.5 実験に用いた合成関数

#### 4.2.5 学習方法

学習方法は実験 1 と同様に図 4.2 のように行った。

#### 4.2.6 評価指標

本実験の評価指標はサンプルデータ数 200 の合成関数において、最初の 1 個のデータを入力したあとに、残りの 199 個のデータを予測した際の予測値と正解データとの誤差である。誤差は各時刻あたりの誤差の絶対値をとり、サンプルデータ数で割り、1 時刻あたりの予測誤差を計算したものをを用いた。



### 4.2.7 実験結果

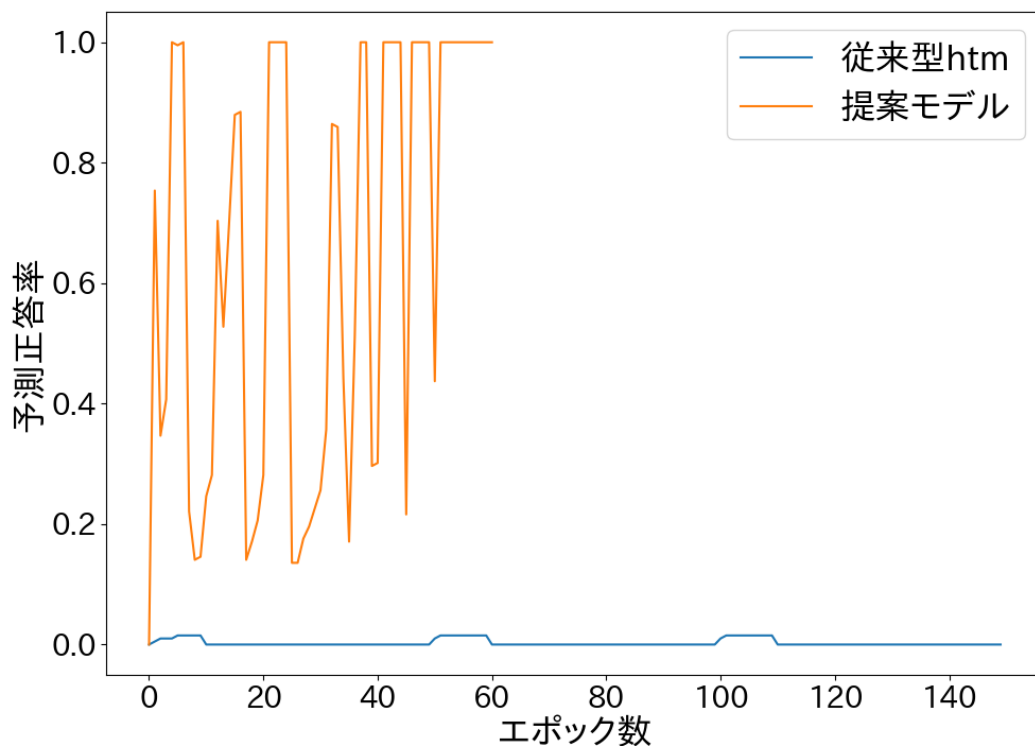


図 4.6 実験 2 の結果

表 4.4 各モデルにおける結果

	提案モデル	従来型 HTM	RNN	LSTM	GRU
予測可能時間	200	28	200	200	200
1 時刻当たりの 予測誤差 (絶対値)	0.0	0.0	0.835	0.560	0.720

提案モデルは従来型 HTM よりも予測正答率が高くなっていることからわかるように予測可能時刻が大幅に改善している。RNN や LSTM、GRU といった機械学習による手法では各時刻において誤差が生じているが、HTM では誤差が生じていない。

以下の図 4.7 4.10 は各モデルにおいて合成関数を再現したものである。

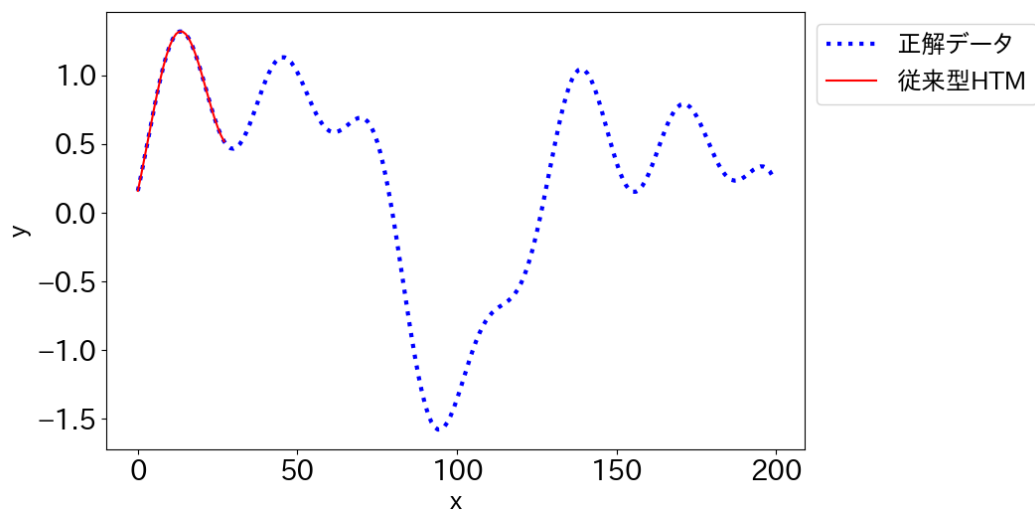


図 4.7 従来型 HTM を用いた合成関数の再現

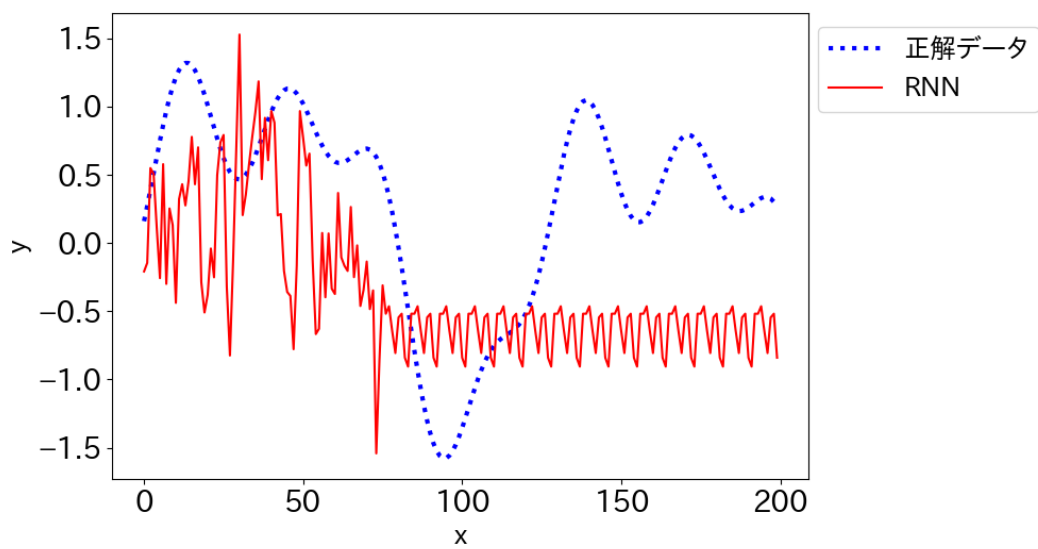


図 4.8 RNN を用いた合成関数の再現

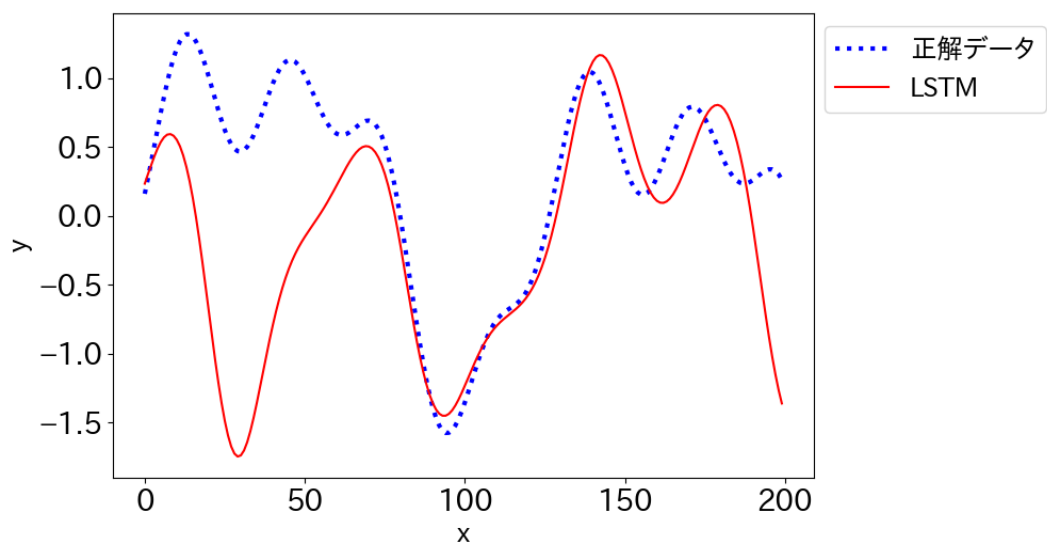


図 4.9 LSTM を用いた合成関数の再現

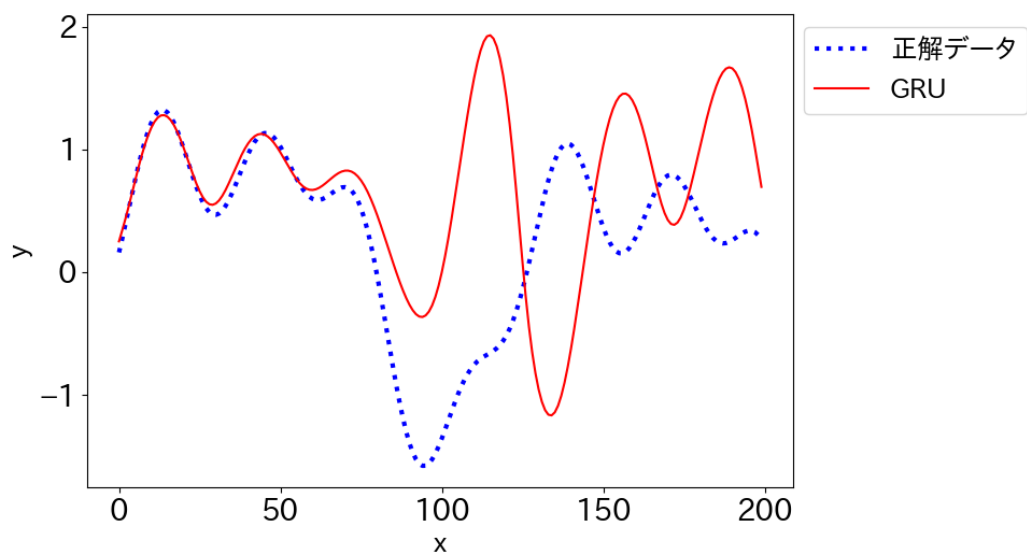


図 4.10 GRU を用いた合成関数の再現

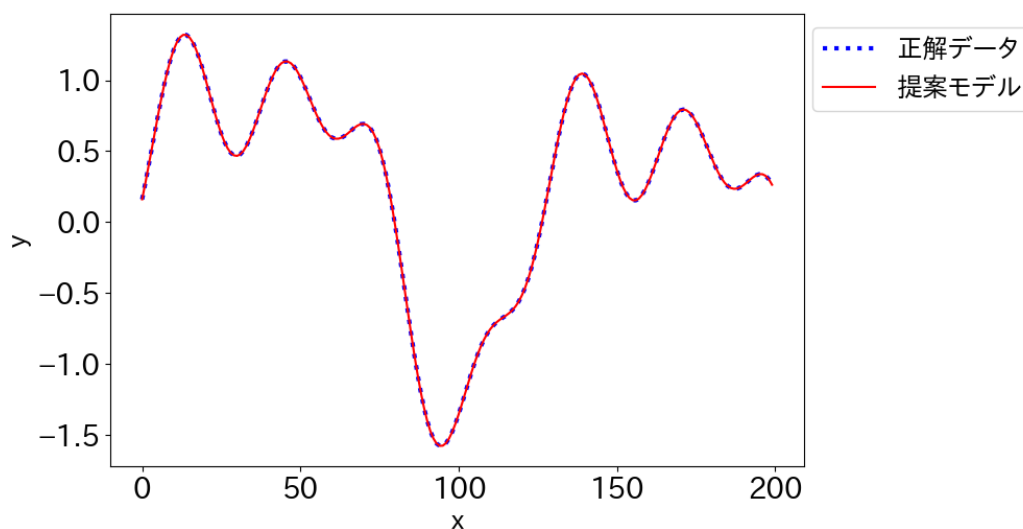


図 4.11 提案モデルを用いた合成関数の再現

#### 4.2.8 実験考察

本実験によって以下の2つの考察が考えられる

1. 提案モデルは長期依存関係の考慮において従来型 HTM の性能を向上させていると言える
2. 提案モデルは機械学習による手法と比較して高い時系列予測精度をもっているため予測誤差がほとんどなくなるような予測ができると言える

1 つ目に関しては、2 章で述べた以下の2つの問題を改善したためだと考えられる。

- a. 疎な分散表現を用いたために発火するセルが徐々に少なくなり消失する。
- b. 学習が大きく進んだパターンにおいて表現が疎になった時に次のパターンに繋がっていたセルが消失するために学習状態が損失する。

特に a の問題において大きい改善があったと考えられる。これは複数の時刻前のデータとの接続を学習し、予測の際にすべての時刻における予測状態のセルを等価に扱ったことによる。これにより予測状態のセルの数が従来型 HTM よ

りも多くなり、かつ予測状態のセルの計算が各時刻に分散することで過学習も防ぐことができた。bの問題は予測精度が最も低かった実験1の条件5の場合においてのみ発生しており、モデルの精度がある程度高い場合においてはbの問題よりもaの問題の方が予測精度に大きく寄与しているものと考えられる。

2つ目に関しては、提案モデルと従来型の両方HTMの予測精度が高いために誤差が0になっていると考えられる。実験1の条件3以外のモデルでは誤予測があったためHTMにおいてもモデルの精度が低いときに大きく予測が異なる場合があると考えられる。また今回は各データをランダムなカラムの組み合わせに変換しているため、連続した値でもHTMも表現上では大きく異なったものとなる。これによって誤予測が起きた場合にその誤差が大きくなるという問題がある。この問題はHTMのカラムの出力と入力値が近い表現になるように学習する機構を導入することで改善できると思われるため今後の課題としたい。

## 第 5 章

### 結論

本論文では、HTM を複数前の時刻におけるパターンを表現するセルとの接続も持つように改良することを提案した。HTM の構造の面では接続セグメントに時間軸を導入した。また HTM の学習アルゴリズムの面では、シナプス接続において複数時刻に渡るセルの発火との関係を持つことと予測状態のセルの計算において複数時刻からの繋がり重ね合わせを行うことで、長期依存関係を保持することに成功した。これによって並列予測を要する時系列データに対しての予測タスクにおいて、従来の HTM よりも高い精度を記録することが確認された。

今後の展望として、分散表現を HTM のカラム表現に変換する方法を確立することと、単語分散表現を用いることで自然言語処理における様々な言語モデルに適応することを考えている。これによって脳の言語処理における語彙を司る分野と文脈を司る分野に分けた学習を模した言語学習が可能になると考えられる。

## 謝辞

本研究を行うにあたり、指導教官の萩原将文教授から終始熱心なご指導を承りました。ここに感謝の意を表します。また研究室の方々には様々な相談をさせて頂き、特に武内先輩、米倉先輩、和田先輩には研究を通じて活発な議論にお付き合い頂きましたことを感謝致します。

## 第 6 章

### 参考文献



## 参考文献

- [1] F. Cummins F.A. Gers J. Schmidhuber. “Learning to forget: continual prediction with LSTM”. In: *Neural Comput* 12.10 (2000).
- [2] H. Haas H. Jaeger. “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication”. In: *Science* (2004), pp. 78–80.
- [3] S. J. Koopman J. Durbin. “Time series analysis by state space methods (2nd ed.)” In: *Oxford University Press* (2012).
- [4] S. Ahmad J. Hawkins. “Why neurons have thousands of synapses, A theory of sequence memory in neocortex”. In: *arxiv* (2015).
- [5] Ç. Gülçehre F. Bougares H. Schwenk Y. Bengio K. Cho B. V. Merriënboer. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: *CoRR* abs/1406.1078 (2014). arXiv: 1406.1078. URL: <http://arxiv.org/abs/1406.1078>.
- [6] C. E. Schreiner M. Brosch. “Sequence sensitivity of neurons in cat primary auditory cortex”. In: *Cereb. Cortex* 10.12 (2000), pp. 1155–1167.
- [7] T. J. Sejnowski R. P. Rao. “Predictive learning of temporal sequences in recurrent neocortical circuits”. In: *Proceedings of the Novartis Found. Symp.* 239 (2001), pp. 208–229.
- [8] L. Rabiner. “An introduction to hidden Markov models”. In: *IEEE ASSP Magazine* 3.1 (1986), pp. 4–16.

- 
- [9] J. Peng R.J. Williams. “An efficient gradient-based algorithm for on-line training of recurrent network trajectories”. In: *Neural Computation* 4 (1990), pp. 491–501.
  - [10] N. Tishby S. Fine Y. Singer. “The hierarchical hidden markov model: Analysis and applications”. In: *Machine Learning* 32.1 (1998), pp. 41–62.
  - [11] J. Schmidhuber S. Hochreiter. “Long short-term memory”. In: *Neural Comput* 9 (1997), pp. 1735–1780.
  - [12] J. Hawkins Y. Cui S. Ahmad. “Continuous online sequence learning with an unsupervised neural network model”. In: *MIT Press Journal* 28.11 (2016), pp. 2474–2504.

## 付録 A

### 実験 1 における $\sin$ 波の再現例

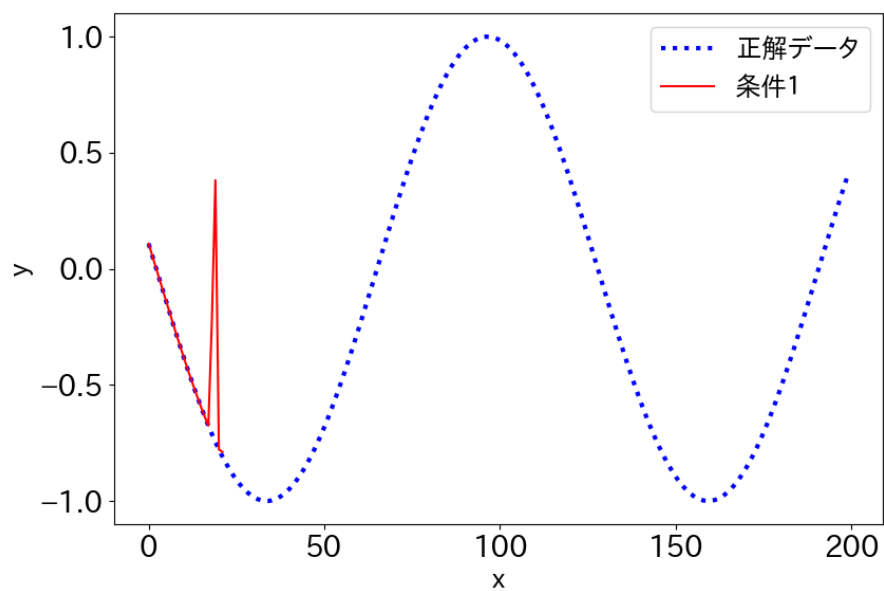


図 A.1 条件 1 における再現例 1

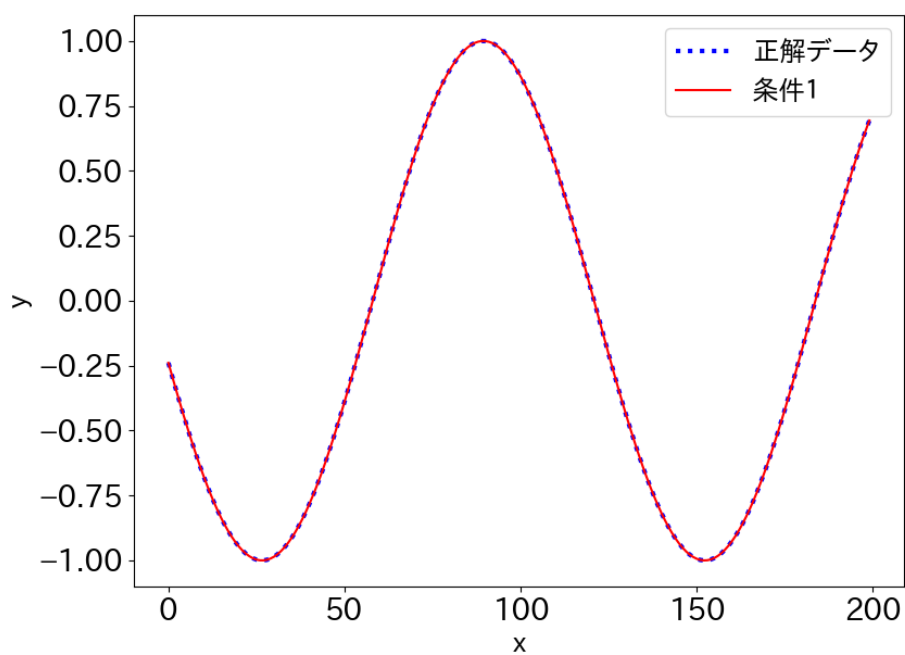


図 A.2 条件 1 における再現例 2

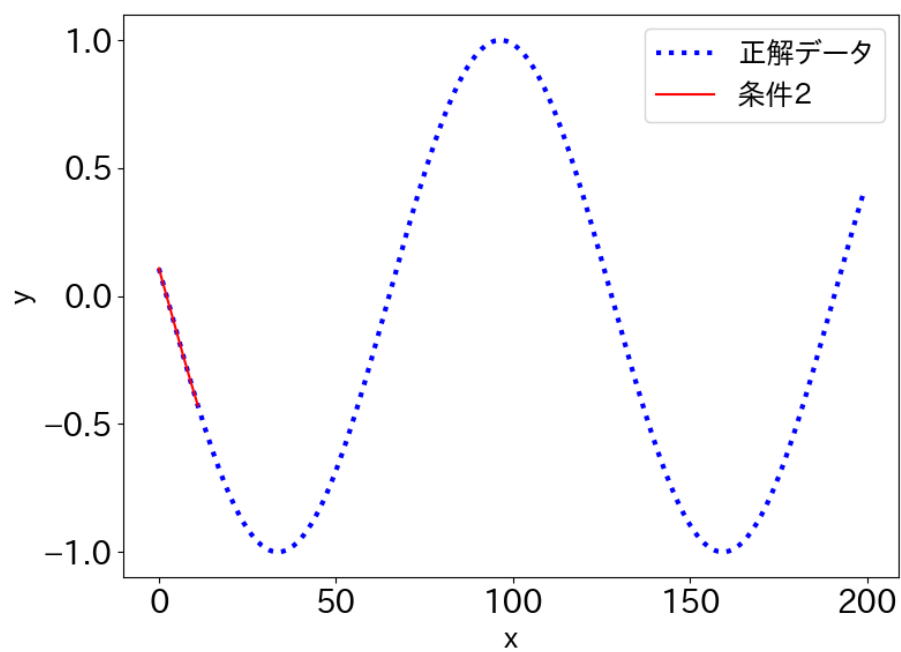


図 A.3 条件 2 における再現例 1

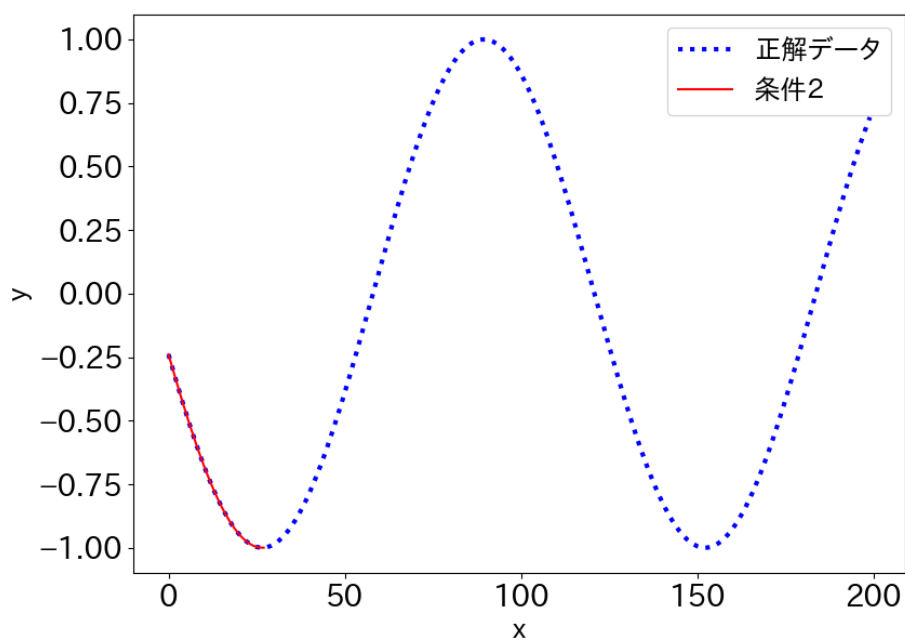


図 A.4 条件 2 における再現例 2

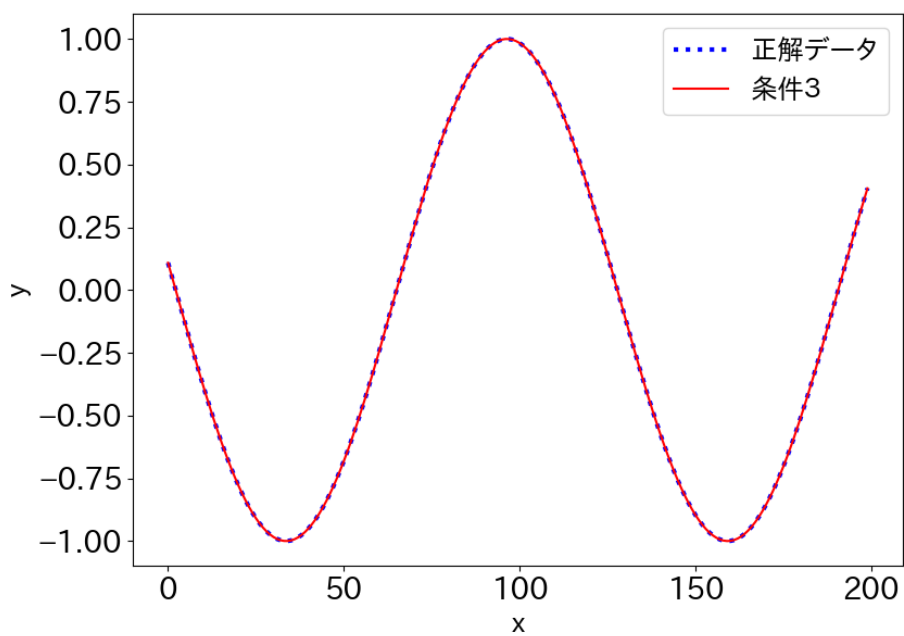


図 A.5 条件 3 における再現例 1

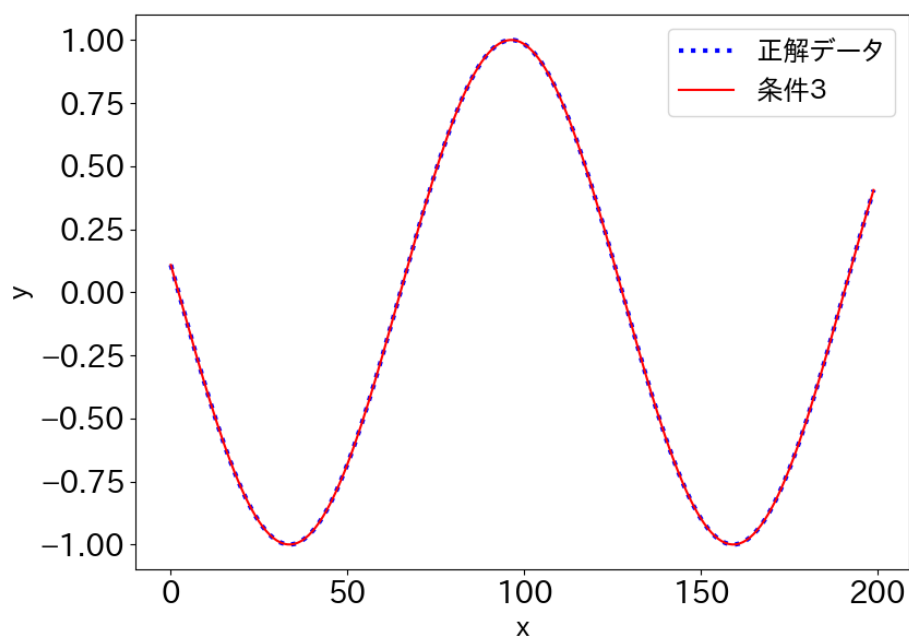


図 A.6 条件 3 における再現例 2

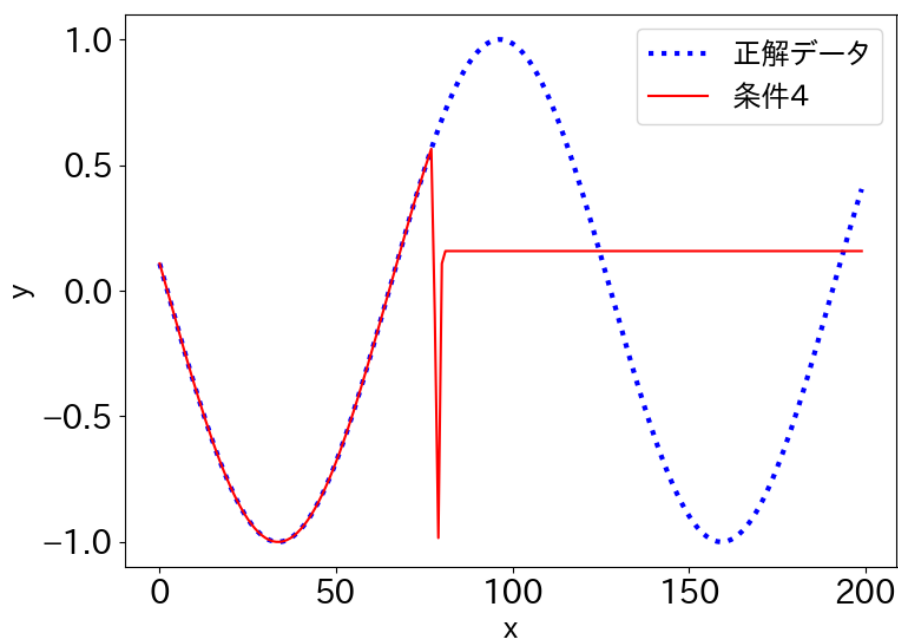


図 A.7 条件 4 における再現例 1

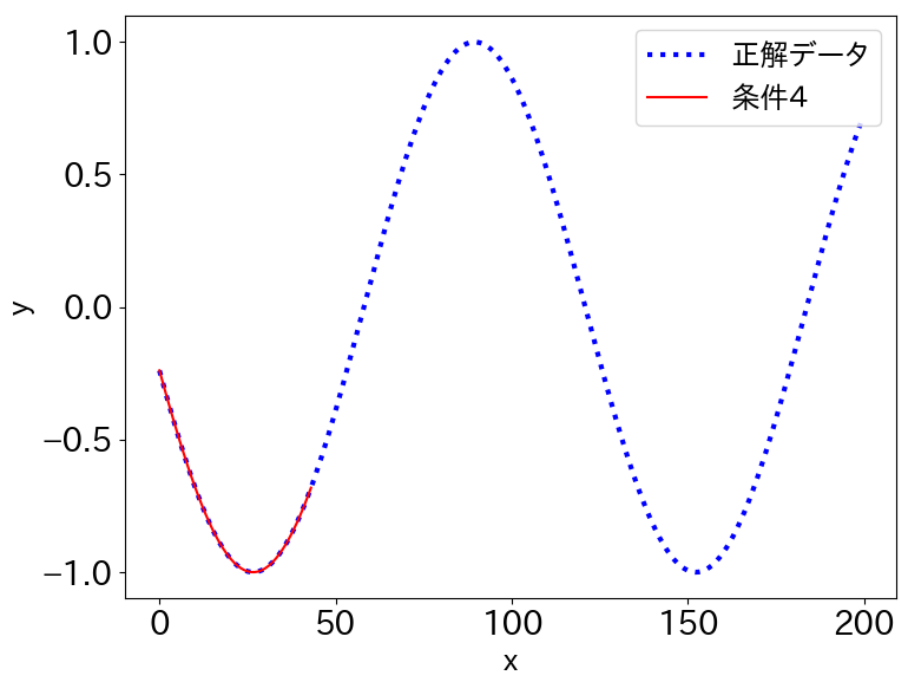


図 A.8 条件 4 における再現例 2

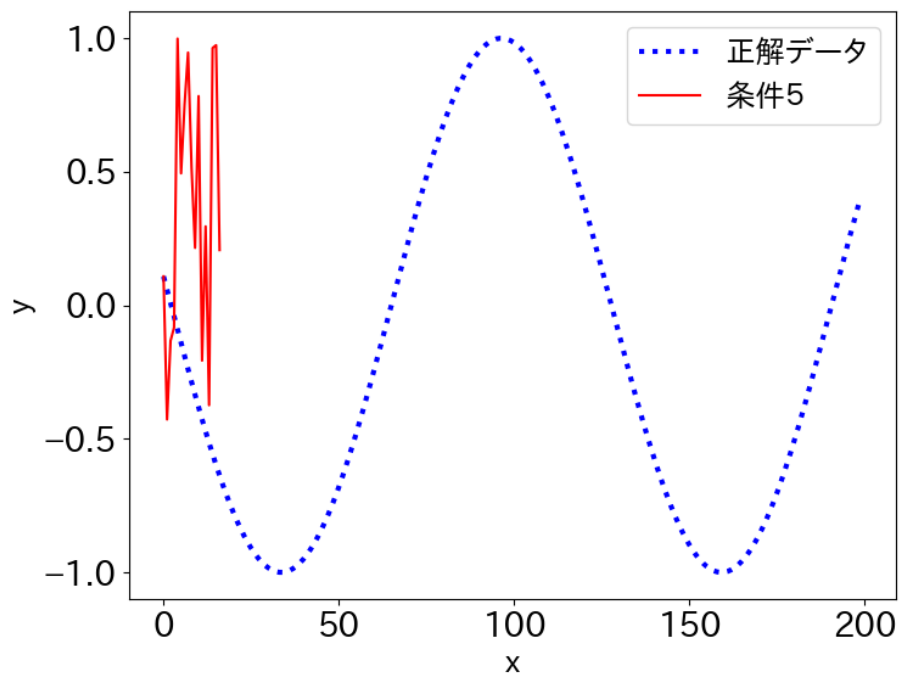


図 A.9 条件 5 における再現例 1

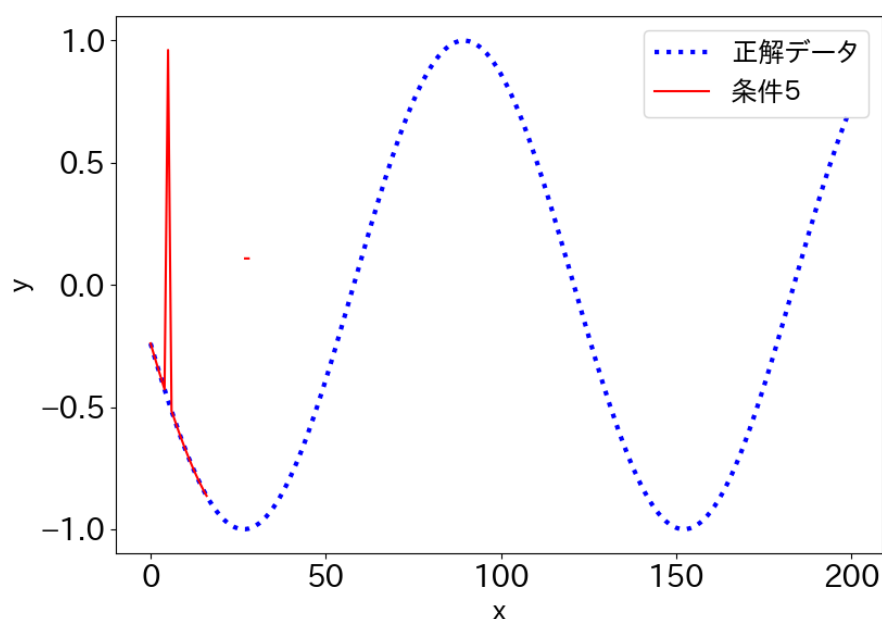


図 A.10 条件 5 における再現例 2

図 A.1 A.10 を見ると条件 3 以外において予測が途中で停止していることが見受けられる。これは予測状態に遷移するセルの量が徐々に減少していき、最終的になくなっているためである。また正解データ以外の値を取るときに大きく離れた値をとっている傾向もある。これは入力値を HTM のカラムの組み合わせの表現に割り当てたときに、入力値上では近い値同士でも HTM のカラムの組み合わせの表現上では大きく離れた値に位置しているためだと考えられる。この改善案として単層ニューラルネットワークなどを用いて入力値の値同士の近さと HTM のカラムの組み合わせの表現における値の値同士の近さが同じになるような学習を加えるというものが考えられる。これを今後の課題としたい。

HTM においてカラムの数は表現力に直接影響するためカラム数を減らした実験 5 では予測が正解データから大きく離れてしまっている。また逆にカラム数が大きくなるほど多くの複雑なデータを扱えるようになるため、今後は実装上の努力によってより大きなカラム数で実行していきたい。



## 付録 B

### HTMの実装

実装時に用いた HTM のパラメータは以下の表の通りである。

表 B.1 HTM のパラメータ	
パラメータ名	値
カラム数	512
1 カラム内のセル数	16
1 セル内の接続セグメント数	32
予測状態に遷移するのに必要な接続セグメント	4
セルの接続値の初期値	0.21
セルにおける接続のしきい値	0.5
接続値を増やす場合の更新式	0.1
接続値を減らす場合の更新式	0.1
予測状態にあったが活性化しなかったセルにおける更新式	0.01
1 ステップ内で新たに持てる接続値の最大数	32

提案モデルと従来型 HTM の両方において python によるスクラッチ実装を行った。ライブラリは numpy と pytorch のみで pytorch は SparseTensor のみを用いた。HTM の活性化状態のセルと予測状態のセルはともに疎な分散表現となるため SparseTensor を用いて実装した。また活性化状態のセルと予測状態のセルからカラム単位での活性化状態の遷移を記録し、それを用いて学習の分岐を行った。セグメント集合は従来型は 5 次元、提案モデルは 6 次元のテンソルを正規分布によってランダムに初期化して用いた。

## 付録 C

### 追加実験