

—論文題目—

時間軸セグメントを導入した 階層的時間記憶モデル

Hierarchical Temporal Memory With
Time Axis Segment Introduced

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	12
2.4.4 並列同時予測	13
2.5 エンコーダと分類器	14
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
3.3.3	セグメント集合の大きさ	19
第 4 章	評価実験	21
4.1	実験 1	21
4.1.1	実験目的	21
4.1.2	実験概要	21
4.1.3	実験条件	21
4.1.4	実験結果	22
4.2	実験 2	22
4.2.1	実験目的	22
4.2.2	実験概要	22
4.2.3	実験結果	22
4.3	実験 3	22
4.3.1	実験目的	22
4.3.2	実験概要	23
4.3.3	実験結果	23
第 5 章	結論	24
	謝辞	24
	参考文献	25
	付録	27
	付録 A HTM のパラメータ	27

付録 A	HTM のパラメータ	27
A.1	従来型の HTM	27

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) に長期依存考慮に関する改良を提案する。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、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、本研究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。そのために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モデルを提案した。よって提案モデルは様々な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 [5] を時系列データの予測に用いたもの [7] 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[6] 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した。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た。その中で LSTM[8] 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を防いだ。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。[9]

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が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があるが、これは神経細胞の樹状突起を模したものである。またこのセグメントは 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。

また 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。

他の HTM 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わずかな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が加えられ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よ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した。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。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。

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。

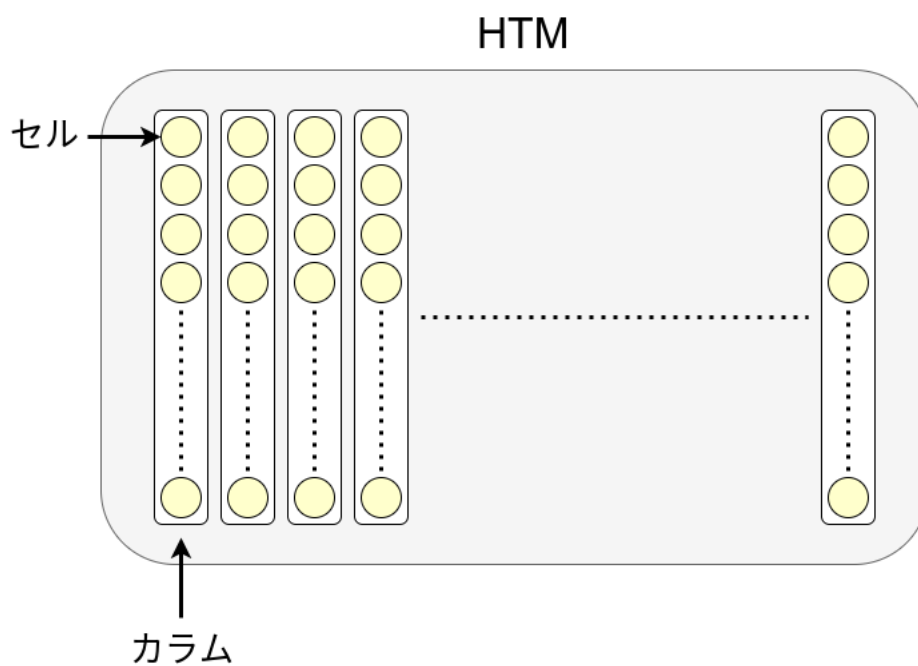


図 2.1 HTM の全体構造

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。

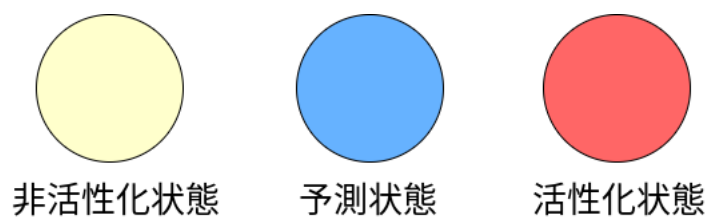


図 2.2 セルの状態

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。

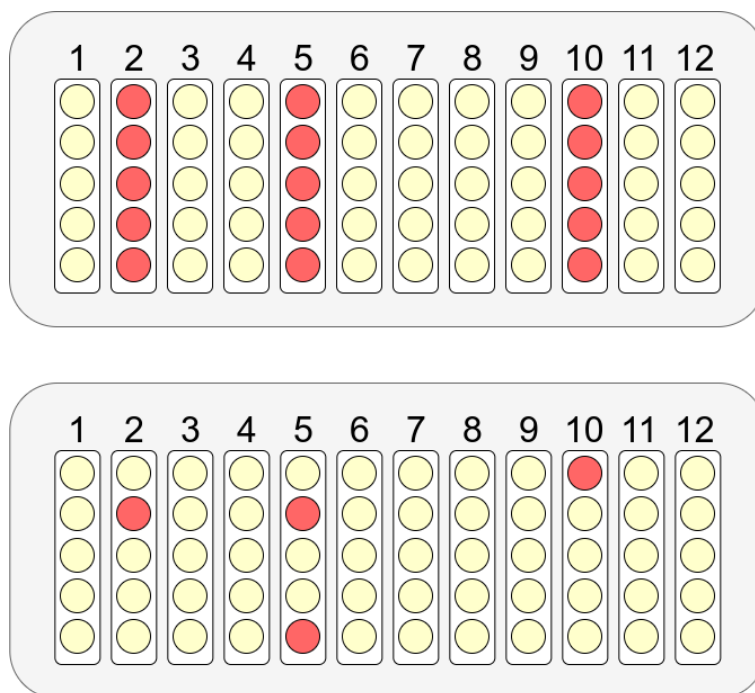


図 2.3 パターンの表現：上下どちらの図も同じカラムの組み合わせ (2, 5, 10) でパターンを表現しているため、両方同じパターンを表す。

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2つからなり、入力セグメントとセグメント集合からなる。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。

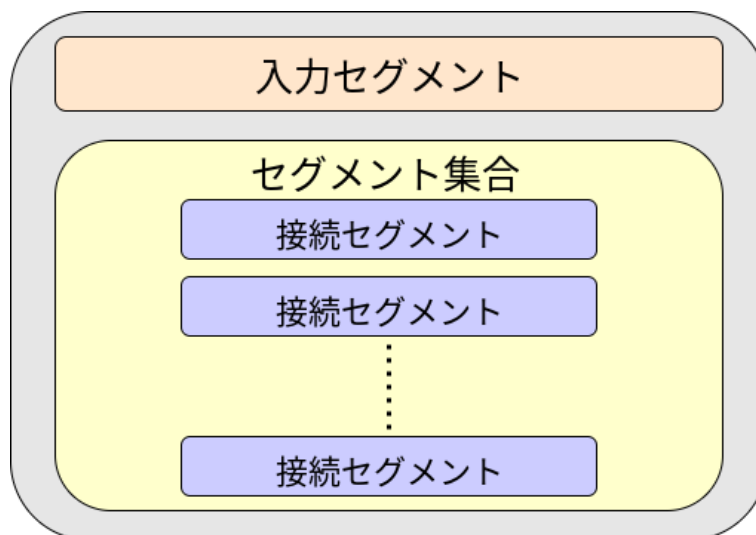


図 2.4 セル内の構造

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。これによ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。

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。同様に予測状態にあるセ

ルの集合を $M \times N$ の 2 進行列 $\mathbf{\Pi}^t$ 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 π_{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 で表させる。

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

この式はセルが予測状態に遷移する場合を表している。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。

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

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。

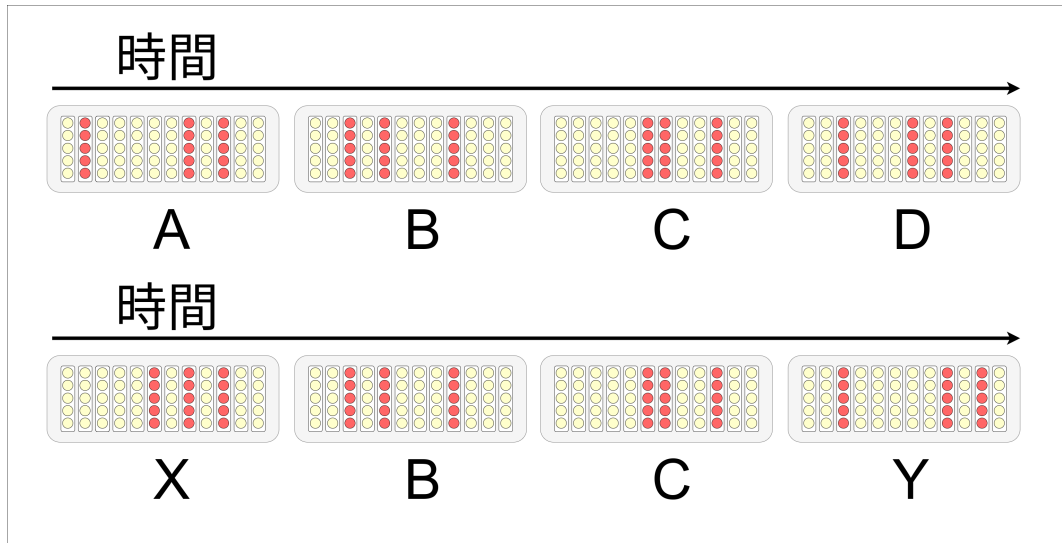


図 2.5 学習前の HTM : (A, B, C, D) という系列と (X, B, C, Y) という系列を入力した場合

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。

2.4.2 学習中

例として図 2.6 を考える。

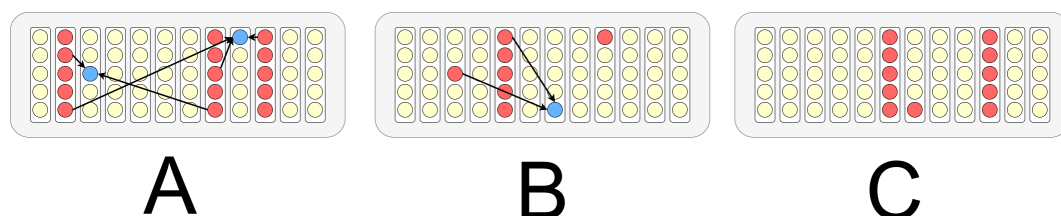


図 2.6 学習中の HTM

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。

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。

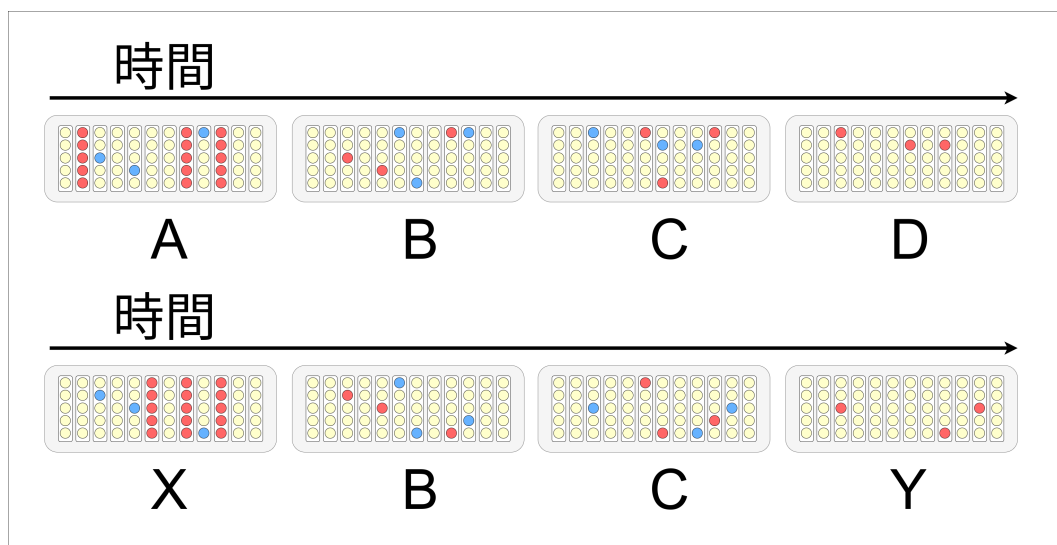


図 2.7 学習後の HTM : (A, B, C, D) という系列と (X, B, C, Y) という系列を入力した場合

2.4.4 並列同時予測

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。

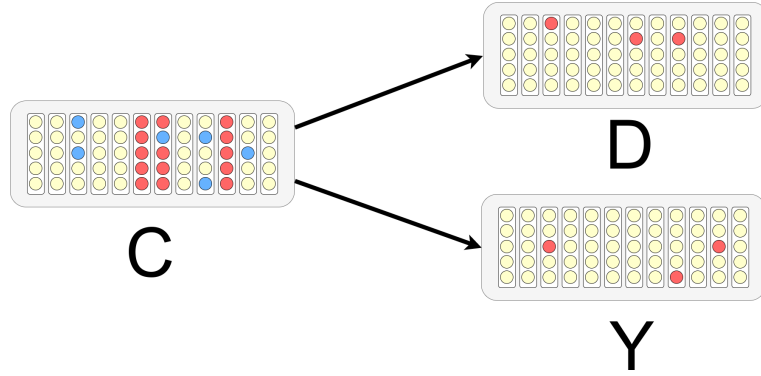


図 2.8 学習後の HTM : (A, B, C, D) という系列と (X, B, C, Y) という系列を学習後に (C) という系列を入力した場合

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せる。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。

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

\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2つの問題点がある。それは以下の2つである。

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。しかし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ときは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ため予測状態になるセルが疎になりすぎてしまい、予測を進める内に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のほとんどのカラム中のセルすべてが活性化する。それに対して、学習が進むにつれて勝者カラムの多くのカラムで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を要する予測タスクに対して脆弱性があった。そのため HTM の性能を改善するために HTM に時間軸セグメント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セルの状態遷移が安定して行われるために、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。

具体的に前章で挙げた以下の 2 つの問題点に関する改良について述べる。

- 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。
- 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が損失する。

まず 1 つ目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 つ目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

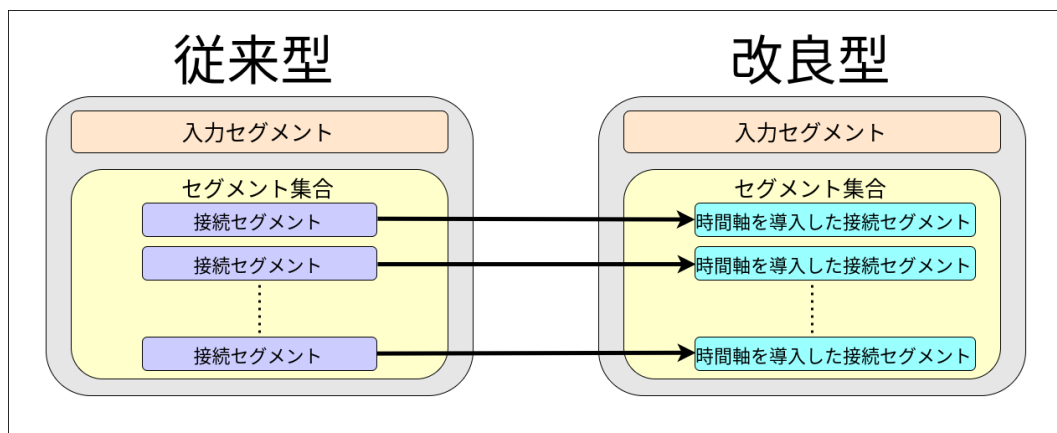


図 3.1 提案モデルにおけるセル内の構造

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 次元テンソル値となる。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した部分がある。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点がある。

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 τ_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た接続の可否のみを表す 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。

3.3.3 セグメント集合の大きさ

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。

第 4 章

評価実験

ここはあとで綺麗に書きます。

4.1 実験 1

4.1.1 実験目的

適切なセグメント集合のサイズ検定

4.1.2 実験概要

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。

4.1.3 実験条件

セグメント集合のサイズ [カラム数、セル数、セグメント数、時間軸長] セグメント集合のテンソルの合計の大きさは (カラム数²*セル数²*セグメント数*時間軸) 従来型 *HTM* のテンソルの合計サイズ (512, 16, 32, 1) の場合 $512^2 * 16^2 * 32 * 1 = 2^{31}$

- 512,16,4,8
- 512,16,8,4

- 512,8,16,8
- 512,8,8,16

4.1.4 実験結果

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定

4.2.2 実験概要

様々な合成関数を学習させた提案モデルを従来型の HTM と LSTM、GRU を比較

4.2.3 実験結果

- 改良型 HTM
- HTM
- LSTM
- GRU

4.3 実験 3

4.3.1 実験目的

改良型 HTM における同時並行予測を必要とする長期依存考慮の性能検定

4.3.2 実験概要

様々な合成関数を学習させた提案モデルを従来型の HTM と LSTM、GRU を比較

4.3.3 実験結果

- 改良型 HTM
- HTM
- LSTM
- GRU

第 5 章

結論

本論文では、時間軸セグメントを導入した HTM を提案した。HTM の構造の面ではセグメント集合の次元を増やし時間軸次元を導入した。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の確立することと、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。

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] L. Rabiner. “An introduction to hidden Markov models”. In: *IEEE ASSP Magazine* 3.1 (1986), pp. 4–16.
- [6] 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.
- [7] N. Tishby S. Fine Y. Singer. “The hierarchical hidden markov model: Analysis and applications”. In: *Machine Learning* 32.1 (1998), pp. 41–62.
- [8] J. Schmidhuber S. Hochreiter. “Long short-term memory”. In: *Neural Comput* 9 (1997), pp. 1735–1780.
- [9] 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

HTMのパラメータ

A.1 従来型の HTM