

# HIERARCHICAL TEMPORAL MEMORY

including

## HTM Cortical Learning Algorithms

**VERSION 0.2, DECEMBER 10, 2010**

©Numenta, Inc. 2010

Use of Numenta's software and intellectual property, including the ideas contained in this document, are free for non-commercial research purposes. For details, see

<http://www.numenta.com/software-overview/licensing.php>

**翻訳 株式会社アルトーク 2011/1/25**

出典: [http://www.numenta.com/htm-overview/education/HTM\\_CorticalLearningAlgorithms.pdf](http://www.numenta.com/htm-overview/education/HTM_CorticalLearningAlgorithms.pdf)

脚注はすべて訳者による注釈である。本書の改訂版を確認するには <http://ai.altalk.com> 参照。

## Numenta 翻訳ライセンス（参考和訳）

---

Copyright (c) 2010, 2011 Numenta, Inc.

All rights reserved.

ここに含まれる文章、アルゴリズム、サンプルコード、擬似コード、及びその他の記述は、Numenta Inc.が発行した hierarchical temporal memory (“HTM”) 技術に関する記述の翻訳ないしこれに基づいて得られたものである。原著の著作権及びここで翻訳された HTM やそのアルゴリズムに関する特許権は Numenta が保有している。独立した HTM システムの開発及び使用に関して、それが商用目的ないし製品化目的ではなく、研究目的である限り、Numenta はその特許権を主張しないことに同意する。Numenta の特許権に抵触する商用目的ないし製品化目的のいかなる HTM 技術の使用も、Numenta から商用ライセンスを取得しなければならない。

上記に基づいて Numenta は貴方に、商用目的ないし製品化目的の使用ではなく、研究目的に限り、これらのアルゴリズム及び著作を使用することを認可する。前述の「商用目的ないし製品化目的の使用」は、訓練された HTM ネットワークないしアプリケーションを後に商用目的ないし製品化目的で適用することを意図している場合、HTM ネットワークを訓練することを含む。前述の「商用目的ないし製品化目的の使用」はまた、商用目的ないし製品化目的で HTM 技術の出力結果を使用ないし他者に使用を許可することを含む。この記述を頒布・出版・複製したあらゆる記述には、この翻訳ライセンスの全文が英文及び翻訳対象言語の両方で含まれていなければならない。

このライセンスは明示的にも暗黙的にも特許権の使用を何ら許可しない。ここで許可された翻訳物の品質ないし正確さに関して Numenta は義務も責任も負わない。

## Numenta Translation License

---

Copyright (c) 2010, 2011 Numenta, Inc.

All rights reserved.

The text, algorithms, sample code, pseudo code and other work included herein are based upon or translated from certain works related to hierarchical temporal memory (“HTM”) technology published by Numenta Inc. Numenta holds the copyright in the original works and patent rights related to HTM and the algorithms translated herein. Numenta has agreed not to assert its patent rights against development or use of independent HTM systems, as long as such development or use is for research purposes only, and not for any commercial or production use. Any commercial or production use of HTM technology that infringes on Numenta’s patents will require a commercial license from Numenta.

Based on the foregoing, Numenta grants you a license to use these algorithms and works for research purposes only and not for any commercial or production use. For purposes of this license, "commercial or production use" includes training an HTM network with the intent of later deploying the trained network or application for commercial or production purposes, and using or permitting others to use the output from HTM technology for commercial or production purposes. Any distribution, publication, or copying of this work must include the full text of this Translation License in both English and the target language.

NO EXPRESS OR IMPLIED LICENSES TO ANY PATENT RIGHTS ARE GRANTED BY THIS LICENSE. NUMENTA SPECIFICALLY DISCLAIMS ANY LIABILITY OR RESPONSIBILITY FOR THE QUALITY OR ACCURACY OF ANY TRANSLATIONS LICENSED HEREUNDER.

## **最初にお読み下さい！**

本書は、この資料のドラフトである。読者が知っておくべき事柄のうち、欠落しているものがいくつかある。

## **本書に含まれること：**

本書は Numenta が 2010 年に開発した学習と予測に関する新しいアルゴリズムの詳細を説明している。この新しいアルゴリズムについて、プログラマが理解でき、もし望むならこれを実装可能なほど十分詳細に説明している。最初の章で概念説明をしている。もし読者が Numenta についてよく知っていて、我々のこれまでの論文のいくつかを読んだことがあるなら、それらは馴染み深いであろう。それ以後の章は新しい事柄について述べる。

## **本書に含まれないこと：**

この新しいアルゴリズムの実装に関するいくつかの話題は、この初期の草稿に含まれていない。

- アルゴリズムの多くの側面は実装及びテストされているが、テスト結果については述べられていない。
- アルゴリズムを実際の問題にどのように適用可能であるかについての記述はない。センサーないしデータベースからのデータを、このアルゴリズムに適した分散表現に変換する方法の記述が抜けている。
- アルゴリズムはオンライン学習ができる。オンライン学習を完全に実装するために、ある特殊な状況下で必要となるいくつかの詳細は記述されていない。
- 執筆予定のその他の議論として次のものがある。疎分散表現の特徴に関する議論、利用例・応用例、付録への引用。

我々は現時点で紹介可能な範囲でこの資料を作成した。他の人々もこれに関心を持つだろうと考えたためである。意欲のある研究者であれば、この資料の欠落している部分は、アルゴリズムを理解し実験をする上で妨げにならないだろうと考える。我々は今後の進展に伴って随時この資料を改訂する。

## 目次

---

Numenta 翻訳ライセンス（参考和訳） .....	2
Numenta Translation License.....	3
序文 .....	8
想定読者 .....	8
ソフトウェアのリリース .....	9
以前の文書との関係 .....	9
Numenta 社について .....	9
著者について .....	10
改訂記録 .....	10
第 1 章： HTM 概説 .....	11
HTM 原論 .....	12
階層構造 .....	12
リージョン .....	15
疎分散表現 .....	16
時間の役割 .....	17
学習 .....	19
推論 .....	20
予測 .....	21
行動 .....	23
HTM の実装に向けての進捗状況 .....	23
第 2 章： HTM 大脳皮質性学習アルゴリズム .....	24
用語説明 .....	24
セル状態 .....	24
樹状突起セグメント .....	25
シナプス .....	25
概要 .....	26
共通概念 .....	32
二値ウェイト .....	32
永続値 .....	33
樹状突起セグメント .....	33
シナプス候補 .....	33
学習 .....	33

空間プーリングの概念.....	34
空間プーリングの詳細.....	36
時間プーリングの概念.....	36
時間プーリングの詳細.....	38
一次と可変長のシーケンスと予測.....	39
第3章： 空間プーリングの実装と疑似コード.....	43
初期化.....	43
フェーズ 1: オーバラップ.....	44
フェーズ 2: 抑制.....	44
フェーズ 3: 学習.....	44
データ構造と補助関数.....	46
第4章： 時間プーリングの実装と疑似コード.....	48
時間プーリング疑似コード： 推論だけのバージョン.....	48
フェーズ 1.....	48
フェーズ 2.....	49
時間プーリング疑似コード： 推論と学習を含むバージョン.....	49
フェーズ 1.....	49
フェーズ 2.....	51
フェーズ 3.....	51
実装の詳細と用語説明.....	52
付録 A: 生体ニューロンと HTM セルの比較.....	57
生体ニューロン.....	57
細胞体.....	58
主要樹状突起.....	58
末梢樹状突起.....	58
シナプス.....	59
ニューロンの出力.....	60
単純な人工ニューロン.....	60
HTM セル.....	61
主要樹状突起.....	61
末梢樹状突起.....	62
シナプス.....	63
セル出力.....	63
参考文献.....	64
付録 B: 新皮質の層と HTM リージョンの比較.....	65
新皮質の神経回路網.....	65

層.....	66
リージョンの違いによる層のバリエーション.....	67
カラム.....	68
ミニカラム.....	69
カラム反応の例外.....	70
なぜ層とカラムがあるのか？.....	71
異なる層が何をするのかに関する仮説.....	72
HTM リージョンは新皮質の何に相当するか？.....	75
まとめ.....	75
用語の説明.....	77

## 序文

---

人間には簡単にできて、コンピュータには今のところできないようなことがたくさんある。画像パターン認識、話し言葉の理解、触ることで対象物を理解したり操作すること、複雑な世界を行き来することなどは人間にとっては簡単なことである。ところが数十年にも及ぶ研究にも関わらず、コンピュータ上で実行可能な人らしい行為をするアルゴリズムはわずかしかない。

人の場合、これらの能力は主に新皮質によって成される。**Hierarchical Temporal Memory (HTM)** は、新皮質がこの様な機能を発揮する様子をモデル化する技術である。HTM は人間と同等あるいはそれ以上の多くの認識性能を持つ機械を製造可能とすることを約束する。

本書は HTM 技術について述べる。第 1 章は HTM を幅広く概説する。階層構造の重要性、疎分散表現<sup>1</sup>、時間的な変化に基づく学習について述べる。第 2 章は HTM 大脳皮質性学習アルゴリズム<sup>2</sup>について詳細に述べる。第 3 章と第 4 章は HTM 学習アルゴリズムの疑似コードを、空間プーリング及び時間プーリングの 2 つのパートに分けて説明する。第 2 章から第 4 章を読めば、熟練したソフトウェア技術者であればこの技術を実装して実験することができるだろう。我々の成果のさらに先を行き、あるいは拡張を試みる読者が現れることを望んでいる。

## 想定読者

本書は技術教育を受けた読者を想定している。ニューロサイエンスに関する予備知識を必要とはしていないものの、読者が数学やコンピュータ科学の概念を理解できることを仮定している。我々は、この文書が授業の教科書として利用可能なように執筆した。最も想定される読者はコンピュータ科学や認知科学の学生、あるいは人の脳と同様の原理で機能する人工認知システムを作成するソフトウェア開発者である。

本書のいくつかの部分、特に第 1 章の HTM 概説は技術者でない読者にも役立つ。

---

<sup>1</sup> sparse distributed representation。本書を理解する上で重要な概念だが、冒頭で述べられているように、残念ながら本書では説明されていない。理論的基礎は Pentti Kanerva 著 Sparse Distributed Memory に詳しい。Kanerva 氏は Jeff Hawkins 氏が設立した Redwood Neuroscience Institute (現在は Redwood Center for Theoretical Neuroscience) の研究員。

<sup>2</sup> HTM cortical learning algorithms

## ソフトウェアのリリース

本書に述べられたアルゴリズムに基づくソフトウェアリリースは 2011 年中頃を予定している。

## 以前の文書との関係

HTM 理論の一部は 2004 年の *On Intelligence*<sup>3</sup>、Numenta 社から発行されたホワイトペーパー、Numenta の従業員が執筆した論文などで述べている。読者がこれらの書類を読んでいることは前提としていない。これらの多くは本書でカバーされ、更新されている。第 2 章から第 4 章で述べている HTM 学習アルゴリズムは、これまで発表されたことはない。この新しいアルゴリズムは、Zeta 1 と呼ばれていた我々の第一世代アルゴリズムを置き換えるものである。初期の頃はこの新しいアルゴリズムのことを“Fixed-density Distributed Representations” ないし “FDR” と呼んでいたが、我々はやこの用語を使用していない。新しいアルゴリズムは HTM 大脳皮質性学習アルゴリズム<sup>4</sup>、あるいは単に HTM 学習アルゴリズムと呼んでいる。

我々は、Numenta 社創設者の一人である Jeff Hawkins と Sandra Blakeslee によって書かれた *On Intelligence* を読まれることを推奨する。この本は HTM という名前で述べてはいないものの、それは HTM 理論とその背景にあるニューロサイエンスについて、読みやすくかつあまり技術よりにならずに説明している。*On Intelligence* が執筆された当時、我々は HTM の基本原理を理解していたが、これをコンピュータアルゴリズムとして実装する方法を知らなかった。本書は *On Intelligence* から始まった研究の続きと考えることができる。

## Numenta 社について

Numenta, Inc. ([www.numenta.com](http://www.numenta.com)) は HTM 技術を商業的ないし学術的利用のために開発することを目的として 2005 年に設立された。この目的を達成するため、我々は我々の進捗及び発見を完全に文書化している。我々はまた、我々が開発したソフトウェアを他の人が研究目的や商業目的で利用できるように提供している。我々は、独立系のアプリケーション開発コミュニティが立ち上がることを支援できるように、このソフトウェアを構築した。Numenta 社のソフトウェアや知的所有権を研究目的で自由に利用

<sup>3</sup> 邦訳「考える脳 考えるコンピューター」ランダムハウス講談社

<sup>4</sup> HTM Cortical Learning Algorithms

してよい。我々は商業目的での技術サポートの提供、ソフトウェアライセンス販売、知的所有権のライセンス販売で収益を得ている。我々は常に開発パートナーを求めており、彼らと我々自身の成功を追い求めている。

Numenta 社はカリフォルニア州 Redwood City に拠点を置き、自己資本で運営されている。

## 著者について

本書は Numenta 社の従業員の協力と努力によって執筆された。各章の主な著者の名前は改訂記録に記載した。

## 改訂記録

各版の主な変更点を下表に示す。細かな修正や整形などは記載していない。

版	日付	変更点	主な著者
0.1	2010/11/9	1. 序文、1,2,3,4 章、用語集の初版	Jeff Hawkins, Subutai Ahmad, Donna Dubinsky
0.1.1	2010/11/23	1. 第 1 章のリージョンの節で、レベル、カラム、層などの用語を明確化するため編集した 2. 付録Aの初版	Jeff Hawkins
0.2	2010/12/10	1. 第 2 章：明確化のため修正 2. 第 4 章：行番号を修正、37行と39行のコードを修正 3. 付録Bの初版	Hawkins Ahmad  Hawkins

## 第1章： HTM 概説

---

Hierarchical Temporal Memory (HTM) は、新皮質の構造的・アルゴリズム的性質を捉えることを目指した機械学習技術である。

新皮質は哺乳類の脳にある、知的思考の中枢である。ハイレベルな視覚・聴覚・触覚・運動・言語・行動計画はすべて新皮質で行われる。そのような様々な認知機能があると聞くと、新皮質がそれぞれに対応して特化されたニューラル・アルゴリズムを実装していると読者は考えるかも知れない。しかしそうではない。新皮質は実に均一なパターンのニューラル回路である。生物学的な証拠によれば、新皮質は多くの異なった知的機能を実行する共通のアルゴリズム一式を実装していることが示されている。

HTM は新皮質及びその多くの能力を理解する理論的な枠組みを提供する。現在我々は、この理論的な枠組みの小さなサブセットを実装している。今後より一層多くの理論が実装されるだろう。現在我々は、新皮質の商業的ないし科学的価値のうちの十分なサブセットを実装したと信じる。

HTM のプログラミングは伝統的なコンピュータプログラミングとは異なる。今日のコンピュータでは、プログラマは特定の問題を解くために特化したプログラムを作成する。これに比べて、HTM はセンサーから得られたデータの流れることで訓練される。HTM の能力はそれがどのようなデータに触れたかによって主に決められる。

HTM はニューラルネットワークの一類型と考えられる。新皮質の構造的詳細をモデル化しようとするどんなシステムもニューラルネットワークと定義することができる。しかしながら、「ニューラルネットワーク」という用語は非常に多くのシステムで用いられているため、不用意に使えない。HTM がモデル化するニューロン (HTM ではセル<sup>5</sup>と呼ぶ) は、カラム<sup>6</sup>、層<sup>7</sup>、リージョン<sup>8</sup>、階層構造<sup>9</sup>の中に配置される。これらの詳細及び HTM 理論はニューラルネットワークの新しい形態である。

名前が示すように HTM は基本的にメモリベースのシステムである。HTM ネットワークは時間的に変化するたくさんのデータによって訓練され、多くのパターンとシーケンス<sup>10</sup>の蓄積に依存している。データを格納及びアクセスする方法は、一般にプログラマが使用する標準的なモデルとは論理的に異なっている。伝統的なコンピュータメモリはフラットに構成され、それ自体は時間に関する概念を持たない。プログラムはフラッ

---

<sup>5</sup> cell

<sup>6</sup> column

<sup>7</sup> layer

<sup>8</sup> region

<sup>9</sup> hierarchy

<sup>10</sup> sequence。連続して起こる事柄、ないしその順序を意味する。

トなコンピュータメモリ上に任意の種類 of データ構造を実装できる。プログラマは情報がどこにどのような形で格納されるかを制御する。これに比べて、HTM のメモリはもっと制限がある。HTM のメモリは階層的な構造であり、時刻の概念が内在している。情報は常に分散型の様式で保存される。HTM の利用者は階層のサイズを指定し、何に対してそのシステムを訓練するのかを定める。しかし情報がどこにどのように格納されるかは HTM が制御する。

HTM ネットワークは伝統的なコンピュータとかなり違うものの、階層構造、時間と疎分散表現（詳しくは後述する）の主要機能を包含する限り、我々は汎用的なコンピュータを使ってモデル化することができる。我々はいずれ、HTM ネットワークに特化したハードウェアを作ることになると信じている。

本書では我々は HTM の特徴と原理を、人の視覚・触覚・聴覚・言語・行動を例にして表す。これらの例は直感的で容易に感じ取ることができるため便利である。しかしながら、HTM の能力は汎用的であることを心に留めておくことは重要である。それらをレーダーや赤外線のような非人間的センサーのデータに触れさせることも容易であり、あるいは金融マーケットデータ、天候データ、Web アクセスパターン、文字列情報などのような純粋な情報主体のデータに触れさせることもできる。HTM は学習と予測の機械であり、様々な種類の問題に適用可能である。

## HTM 原論

本節では、HTM のいくつかの核となる原論を述べる：なぜ階層的な組織が重要なのか、HTM のリージョンはどのように構成されるか、なぜデータを疎分散表現で格納するのか、なぜ時間ベースの情報がクリティカルであるのか。

### 階層構造

HTM ネットワークは階層的に配置されたリージョンからなる。リージョンは HTM における記憶と予測の主要構成要素であり、次節でより詳しく述べる。通常、各 HTM リージョンは階層構造の 1 レベルを表す。階層構造を上がるに伴って、常に集約<sup>11</sup>がある。子リージョンの複数の要素が親リージョンの一つの要素に集約する。一方、階層構造を下がるに伴って、フィードバック接続による情報の発散<sup>12</sup>がある。（リージョンとレベルはほとんど同義である。リージョンの内部的な機能について述べるときに「リージョン」の用語を用い、特に階層構造の中でのリージョンの役割を指すときに「レベル」の用語を用いる）

---

<sup>11</sup> convergence

<sup>12</sup> divergence

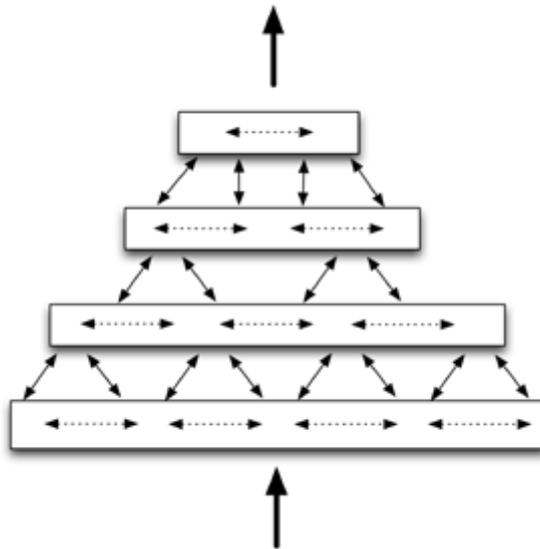


図 1-1 4階層の階層構造に配置された4つのHTMリージョンを単純化して表した図。情報は階層間及び階層内部で通信される。

複数の HTM ネットワークを結合することもできる。このような構造は、1つ以上の情報源やセンサからのデータがあるときに有意義である。例えば、一つのネットワークが音声情報を処理し、他のネットワークが映像情報を処理する場合がある。各個別のネットワークがトップに向かうにつれて集約される。

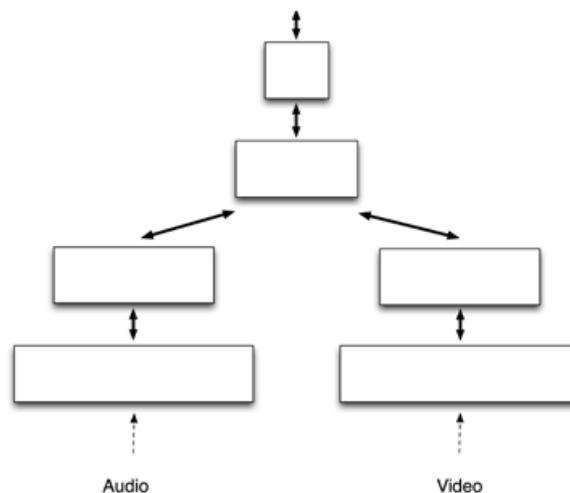


図 1-2 異なるセンサから集約するネットワーク

階層的な構造の利点は効率にある。階層構造の各レベルで学習されたパターンが上位のレベルで組み合わせて再利用されることで、それは学習時間とメモリ消費を非常に節約する。説明のため、視覚について考えてみよう。階層構造の最下位レベルでは、脳は縁<sup>13</sup>や角などの視覚のごく一部分に関する情報を格納する。縁は世の中の様々な物体を構成する基本的な構成要素である。これらの下位レベルのパターンは中間レベルで集約されて、例えば曲線や模様などのより複雑な構成要素に集約される。円弧は耳の縁になりうるし、車のハンドルの上にも、コーヒーカップの取っ手にもなりうる。これらの中間レベルのパターンはさらに集約されて、頭、車、家などの高レベルな物体の特徴を表す。高レベルな物体を学習するとき、その構成要素を再度学習する必要がなくなる。

他の例として、新しい単語を学習するとき、文字や文節、発音を再度学習する必要はない。

階層構造間で表現を共有することはまた、予期される行動の一般化にもなる。もし新しい動物を見たとき、口や歯を見れば、その動物がその口で食べることや、あるいは噛み付く可能性があることを予測できるだろう。階層構造により、世の中の新しい物体がその構成要素が持つ既に分かっている特徴を引き継いでいることを知ることができる。

一つの HTM 階層構造はいくつの事柄を学習できるだろうか？ 言い換えれば、階層構造にはいくつのレベルが必要だろうか？ 各レベルに割り当てるメモリと、必要なレベル数の間にはトレードオフがある。幸い、HTM は入力の実績及び割り当てられたリソースの量とから、各レベルの最適な表現を自動的に学習する。もしあるレベルにより多くのメモリを割り当てたなら、そのレベルはより大きくより複雑な表現を構成し、従って必要となる階層構造のレベルはより少なくなるだろう。もし少ないメモリを割り当てたなら、より小さく単純な表現を構成し、従って必要となる階層構造のレベルはより多くなるだろう。

ここからは、視覚の推論<sup>14</sup>のような難しい問題について述べる（推論はパターン認識と似たものである）。しかし多くの価値ある問題は視覚より単純で、一つの HTM リージョンでも十分であると認められる。例えば我々は、人が Web サイトを眺めていたときに次にどこをクリックするか予測する問題に HTM を適用してみた。この問題は、一連の Web クリックのデータを HTM ネットワークに流し込むことで行った。この問題では空間的階層構造はわずかしか認められない。解決策は主に時間的な統計を見つけることが求められる。即ち、一般的なユーザのパターンを認識することで、ユーザが次にどこをクリックするかを予測する。HTM の時間的学習アルゴリズムはこのような問題に適している。

---

<sup>13</sup> edge。へり・ふち。

<sup>14</sup> inference

まとめると、階層構造は学習時間を節約し、メモリ消費を節約し、一般化をもたらす。しかしながら、単純な予測問題の多くは一つの HTM リージョンでも解決しうる。

## リージョン

階層構造に連結されたリージョン<sup>15</sup>の表現は、生物学からもたらされた。新皮質は厚さ 2mm のニューロンの大きな皮である。生物学では主にそれらが互いにどのように接続しているかに基づいて、新皮質を異なる領域ないしリージョンに区分けする。あるリージョンはセンサから直接入力を受け取るが、他のリージョンは他のいくつかのリージョンを経由してから入力を受け取る。階層構造を決めるのはリージョンからリージョンへの接続関係である。

新皮質のすべてのリージョンの細部は似ているように見える。そのサイズや階層構造の中のどこに位置するかということについての違いはあるものの、その他は似ている。厚さ 2mm の新皮質リージョンを縦にスライスしたなら、6つの層を見ることができる。5つはセルの層で、1つはセルではない層である（少しの例外はあるが、これが一般的な規則である）。新皮質リージョンの各層はカラム状に数多くの相互接続されたセルがある。

HTM リージョンもまた、高度に相互接続されたセルがカラム状に配列された皮からなっている。新皮質の第3層はニューロンの主要なフィード・フォワード層である。HTM リージョンのセルはおおまかに言えば新皮質のリージョンの第3層にあるニューロンと等価である。

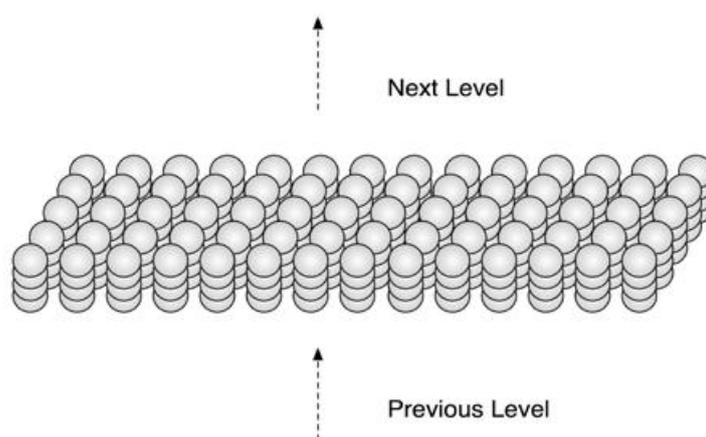


図 1-3 HTM リージョンの区画。HTM リージョンは数多くのセルからなる。セルは二次元のカラム状に配置される。この図では、1カラム当たり4つのセルで構成される HTM リージョンの小さな区画を表している。各カラムは入力

<sup>15</sup> region。体の部位、局部などの意。訳文では単独の region はそのままリージョンとした。

の一部に接続され、各セルは同一リージョン内の他のセルに接続する（接続は図示していない）。この HTM リージョン及びそのカラム構造は新皮質リージョンの一つの層に等価である点に注意。

HTM リージョンは新皮質リージョンのほんの一部と等価であるに過ぎないものの、複雑なデータ列の推論と予測を実行することができ、多くの問題に有益である。

## 疎分散表現

新皮質のニューロンは高度に相互接続しているが、わずかなパーセンテージのニューロンだけが一度にアクティブになるように抑制ニューロンによって保護されている。よって脳内の情報は常に、数多く存在するニューロンのうちのわずかなパーセンテージのアクティブなニューロンによって表されている。この様なコード化は「疎分散表現」と呼ばれる。「疎」とは、わずかなパーセンテージのニューロンだけが一度にアクティブになることを意味している。「分散」とは、何かを表現するためには多くのニューロンがアクティブになる必要があることを意味している。一つのアクティブなニューロンは何らかの意味表現に関わっているが、いくつかのニューロンの文脈の中で解釈されて初めて完全に意味を成すことができる。

HTM リージョンもまた、疎分散表現を使用している。実際、HTM リージョンの記憶の仕組みは疎分散表現に依存しており、それなしには機能しない。HTM リージョンの入力は常に分散表現であるが、必ずしも疎であるとは限らないので、HTM リージョンが最初に行うべきことは入力を疎分散表現に変換することである。

例えば、あるリージョンが 20,000 ビットの入力を受け取るとする。入力ビットの中の“1”や“0”の割合は、時間と共に非常に頻繁に変化するだろう。あるときは 5,000 個のビットが“1”であったり、またあるときは 9,000 個のビットが“1”であったりする。HTM リージョンはこの入力を 10,000 ビットの内部表現に変換して、入力のうちの何ビットが“1”であろうがその 2%にあたる 200 ビットが一度にアクティブになるようにする。HTM リージョンの入力が時間と共に変化するに従って、内部表現もまた変化するが、10,000 ビットのうち約 200 ビットが常にアクティブになる。

リージョン内で表現可能なものの数よりも起こりうる入力パターンの数の方がずっと大きいから、この処理によって多くの情報が失われるのではないか、と思われるかも知れない。しかしながら、どちらの数も途方もなく大きい。リージョンが入力からどのようにして疎表現を作成するかについては後述する。理論的な情報のロスは、実際上は問題にならない。

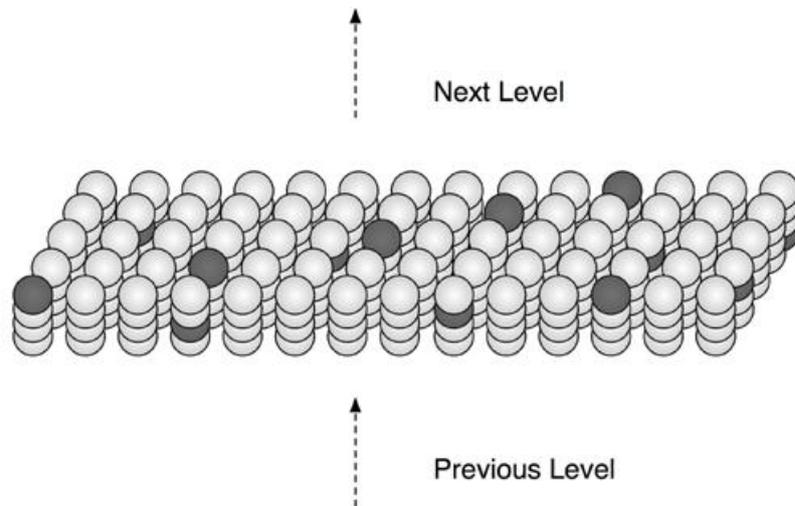


図 1-4 HTM リージョンのセルが疎分散的にアクティブになっている様子

疎分散表現はいくつかの望ましい特徴を有し、HTM の動作になくってはならない。後ほど、この点について再度述べる。

## 時間の役割

時間は、学習・推論・予測において極めて重要な役割を果たす。

推論から始めよう。時間を用いなければ、我々は触覚や聴覚からほとんど何も推論できない。例えば仮に読者が目が不自由だとして、誰かが貴方の手の上にりんごを置いたとしよう。ほんの数秒間触ってみることでそれが何か分かるだろう。りんごの上で指を動かせば、触覚から得られる情報が常に変化しているにも関わらず、その物体そのもの — そのりんごや貴方が持つ「りんご」という高レベルの認識 — は変化しない。しかしながら、もし貴方が手のひらを開いて、その上にりんごが置かれて、しかも手や指先を動かしてはいけないと言われたなら、それがレモンではなくりんごであると識別するのは非常に難しいだろう。

同じことは聴覚についても言える。変化しない音はわずかな意味しか持たない。「りんご」という言葉や、誰かがりんごを噛んだときの音などは、時間と共に素早く順序的に変化する数十から数百の音階の列によってのみ理解しうる。

視覚は対照的に、混在したケースである。触覚や聴覚とは異なり、人は画像が一瞬だけ目の前をすばやく通り過ぎた場合でも識別可能である。よって視覚の推論では必ずしも時間的な入力の変化を必要としない。しかしながら、通常の視覚では我々は常時、目や頭や体を動かしており、物体もまた周囲を動き回っている。素早く変化する視覚的变化の中から推論する我々の能力は、視覚の統計的な特徴と長年の訓練によってもたら

される特別なケースである。視覚・聴覚・触覚における一般的な場合では、推論には時間的に変化する入力が必要である。

推論の一般的なケースと、静的な画像を推論するときの視覚の特別なケースを押さえたところで、学習について見てみよう。学習するには、すべての HTM システムは訓練の間、時間的に変化する入力に触れる必要がある。視覚では静的な画像の推論がときには可能なものの、物体がどのようなものであるかを学習するためにはその物体が変化する様子を見る必要がある。例えば、犬が読者に向かって走ってくる様子を想像してみよう。時間的なそれぞれの瞬間において、犬の画像が貴方の眼の奥の網膜に一連のパターンを形成する。貴方はこれらのパターンを同じ犬の異なる視点を表していると受け止めるが、数学的に言えばそれらのパターンはほとんど似ても似つかない。脳はこれらの異なるパターンが同じものを意味しているということ、その順序的な変化を観察することによって知る。時間はどの空間的なパターンと一緒に現れるかを教えてくれる「先生」である。

センサから得られる入力に変化するだけでは十分ではない点に注意されたい。無関係な入力パターンが続けて現れるても混乱するだけである。時間的に変化する入力は、世界のある固定した情報源からもたらされなければならない。また、我々が人の感覚器官を例として取り上げているものの、非人間的なセンサもまた一般に適用できる点にも注意されたい。もし発電所の温度・振動・雑音のパターンを認識するように HTM を訓練したいのなら、HTM はこれらのセンサの時間的な変化からもたらされるデータで訓練する必要がある。

普通、HTM ネットワークは多くのデータで訓練する必要がある。読者が犬を識別することを学習するには、一匹の犬の一枚の写真ではなく、数多くの種類の犬を何度も見る必要がある。HTM アルゴリズムの仕事は、入力データの時系列の流れを学習することであり、そしてどのパターンに続いて別のどのパターンが現れるかというモデルを構築することにある。この時系列がいつ始まりいつ終わるのかがわからないので、この仕事は難しい。同時に複数の時系列が重なりあって起こることもある。学習は継続的に行われ、またノイズがある中で行われなければならない。

シーケンスの学習と認識は予測を形成する基準となる。どのパターンが他のどのパターンに続くかを HTM が学習すれば、与えられた現在の入力とその直前の入力に対して次にどのパターンが現れる可能性が高いかを予測することができる。予測は後に詳しく述べる。

HTM の 4 つの基本的な機能に戻ろう：学習・推論・予測・行動<sup>16</sup>である。各 HTM リージョンは最初の 3 つの機能、学習・推論・予測を実行する。しかしながら行動は異なる。生物学によれば、多くの新皮質のリージョンが行動を形成することが分かっている。しかし我々は、多くの興味深いアプリケーションにおいてこれは重要ではないと信

---

<sup>16</sup> behavior

じている。よって行動は現在の HTM の実装に含まれていない。ここでは議論の完全性のために触れた。

## 学習

HTM リージョンはセンサから得られるデータのパターンとパターンのシーケンスを見つけることで、その世界を学習する。リージョンはその入力は何を表しているのかを「知って」はいない。それは純粹に統計的な世界でのみ機能する。それは入力ビットの組み合わせのうち、頻繁に同時に起こる組み合わせを見ている。我々はこれを空間的パターンと呼んでいる。そしてこれらの空間的パターンが時間と共にどのような順で現れるかを見ている。我々はこれを時間的パターンないしシーケンスと呼んでいる。

もしリージョンへの入力がある建物の環境に関するセンサであるなら、リージョンは建物の北側や南側において、ある温度と湿度の組み合わせがしばしば起こることを見つけるだろう。そしてこれらの組み合わせが毎日移り変わる様子を学習するだろう。

もしリージョンへの入力があるお店の購入に関する情報であれば、週末にある種の雑誌が購入されることや、天候が寒いときはある種の価格帯のものが夕方頃に好まれることを見つけているだろう。そして異なる人の購入パターンが類似の時系列のパターンに従うことを学習するだろう。

一つの HTM リージョンは学習の能力が限定されている。リージョンはそれがどれだけのメモリを利用可能で、それが受け取った入力がどのくらい複雑であるかに応じて何を学習するかを自動的に調整する。リージョンに割り当てられたメモリが削減されたら、リージョンが学習する空間的パターンはより単純なものとなる。割り当てられたメモリが増加すると、学習する空間的パターンは複雑になりうる。学習した空間的パターンが単純であれば、複雑な画像を理解するにはリージョンの階層構造が必要となりうる。我々はこの特徴を、人の視覚システムに見ることができる。網膜から情報を受け取る新皮質のリージョンは、視覚的な小さな領域についてだけ、空間的なパターンを学習する。階層構造のいくつかのレベルを経由した後だけに、視覚の全体像を認識する。

生物的システムと同様に、HTM リージョンの学習アルゴリズムは「オンライン学習」ができる。即ち、新しい入力を受け取るごとに継続的に学習する。学習した後の方が推論が改善されるが、学習フェーズと推論フェーズとを分ける必要はない。入力のパターンが変化するに従い、HTM リージョンもまた段階的に変化する。

初期の訓練の後、HTM は学習し続けることもできるし、訓練フェーズの後に学習を無効化することもできる。他の方法としては、階層構造の下位レベルでは学習を無効化し、上位レベルでは学習を続けることもできる。HTM がその周囲の世界の統計的構造を学習したら、多くの学習は階層構造のより上位のレベルで起こる。もし HTM が下位レベルではかつて見られなかった新しいパターンに触れたら、これらの新しいパター

ンを学習するのにより長時間必要となる。この特徴は人間にも見られる。既に知っている言語の新しい単語を学習するのは比較的容易である。しかし慣れない発音の外国語の新しい単語を覚えようと思えば、まだ下位レベルの発音を知らないのですと難しいと思うだろう。

単にパターンを見つけることは、価値の高い可能性を秘めている。マーケットの変動、病状の変化、天候、工場の生産、送電系統のような複雑なシステムの障害などの、高レベルなパターンを理解することはそれ自体に価値がある。それでも空間的・時間的パターンを学習することは推論と予測に先立って必要となる。

## 推論

HTM が周囲の世界のパターンを学習すると、新しい入力について推論ができるようになる。HTM が入力を受け取ると、以前に学習した空間的ないし時間的パターンと照合する。新しい入力が以前に格納したシーケンスとうまく適合することが、推論とパターンマッチングの本質である。

メロディをどうやって理解するかを考えてみよう。メロディの最初の音を聞いただけでは良く分からない。次の音を聞けば可能性がかなり狭まるが、まだ十分ではないだろう。メロディを理解するためには普通は3, 4, ないしそれ以上の音を聞く必要がある。HTM リージョンの推論も似ている。それは継続的に入力列を見て、以前学習したシーケンスと照合を試みる。HTM リージョンはシーケンスの最初からの照合を見つけることもできるが普通はもっと流動的で、それはちょうどメロディがどこから始まってでも貴方が理解できることと似ている。HTM リージョンは分散表現を用いるので、リージョンがシーケンスを記憶ないし推論することは上記のメロディの例よりも複雑である。しかしこの例は、HTM が働く様子を示すものである。

貴方がセンサから新しい入力を受け取ったら、それがすぐに明確になるとまでは言えないものの、慣れ親しんだパターンをその新しい入力から容易に見つけることができる。例えば貴方は、例え老人であっても若い人であっても、男でも女でも、早く話されてもゆっくりでも、なまりが強くて、ほとんどの人が話す「朝食」という言葉を理解できる。同じ人が同じ「朝食」という単語を百回発音しても、その音は二度と、貴方の蝸牛殻<sup>17</sup>（音の受容体）を正確に同じように刺激することはないにも関わらずである。

HTM リージョンも脳と同じ問題に直面する：入力は決して正確に繰り返されない。さらに、ちょうど脳と同じように、HTM リージョンは推論や訓練の最中にも新しい入力を取り扱わなければならない。HTM リージョンが新しい入力に対処する一つの方法は、疎分散表現を利用することによる。疎分散表現の鍵となる特徴は、パターンの一部

---

<sup>17</sup> cochleae. かぎゅうかく。耳の奥にある渦巻き状の感覚器官。

分だけをマッチングするだけでほぼ確実にマッチさせることができるということである。

## 予測

HTM の各リージョンはパターンのシーケンスを格納する。格納されたシーケンスを現在の入力とマッチングすることで、次に到着すると思われる入力の予測をする。HTM リージョンは実際には疎分散表現の間の変遷を記録する。あるときはその変遷はメロディの中の音に見られるように線形のシーケンスであるが、一般的な場合は将来入力される可能性があるものが同時に多数予測される。HTM リージョンはときには長期間に及ぶ過去の文脈に基づいて異なる予測をする。HTM の記憶の多くはシーケンスの記憶と空間的パターンの変遷を記憶することに使われる。

HTM の予測の鍵となる特徴のいくつかを以下に示す。

### 1) 予測は継続的である

貴方は特に意識していなくても継続的に予測している。HTM も同じことをする。歌を聞いているとき、貴方は次の音を予測している。階段を降りるとき、貴方の足がいつ次の段に触れるかを予測している。野球のピッチャーの投球を見ている時、ボールがバッターの方に近づいていくことを予測している。HTM リージョンでは、予測と推論はほとんど同じことである。予測は分離された処理ではなく、HTM リージョンが働く仕組みに統合されている。

### 2) 予測は階層構造のすべてのレベルのすべてのリージョンで起こる

HTM リージョンが階層構造を持つとき、予測はすべてのレベルで起こる。リージョンはそれが既に学習したパターンについて予測をする。言語の例では、低レベルのリージョンでは次の音素を予測し、高レベルのリージョンでは単語や句を予測するだろう。

### 3) 予測は文脈依存である

予測は過去に何が起こったか、そして現在何が起きているかに基づいて行われる。従って直前の文脈に基づいて、ある入力から異なった予測が行われることがある。HTM リージョンは必要なだけのより多くの直前の文脈を用いて学習し、短時間ないし長時間の両方の文脈を保持することができる。この特徴は可変長記憶<sup>18</sup>として知られている。例えば、暗唱している演説、ゲティスバーグ演説<sup>19</sup>などを考えてみよう。次の単語を予

<sup>18</sup> variable order memory

<sup>19</sup> Gettysburg Address. 「人民の人民による人民のための政治」のフレーズが有名。"Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal..." (以下略)

測するには、現在の単語だけでは全く不十分である。最初の文章だけでも “and” の次に “seven” が来ることもあれば、“dedicated” が来ることもある。ときにはほんの少しの文脈で予測できることもある。“four score and” と分かれば次の “seven” を予測できる。他の場合はフレーズが繰り返される場合があり、演説のうちのどの部分であるかを知るためには、そして次に何が来るかを予測するには、より長い文脈を使う必要がある。

#### 4) 予測は安定化につながる

あるリージョンの出力はその予測である。HTM の特徴の一つは、リージョンからの出力はより安定しているということである。安定とはつまり、階層構造の上位に行くほどよりゆっくりと変化し、長く継続するという。この特徴はリージョンが予測する方法からもたらされる。リージョンは単にすぐ次に何が起こるかを予測するだけではない。可能なときは時間軸の複数ステップ先を予測する。あるリージョンが 5 ステップ先を予測できるとしよう。新しい入力に到着したとき、新たに予測されたステップは変化するが、しかしそのうちの 4 つの既に予測されたステップは変化しない。従って、個々の新しい入力は全く違っていても、出力の一部だけが変化するので、出力は入力よりも安定化している。この特徴は我々が実世界で経験することを反映している。歌の名前のような高レベルの概念は、歌の中の実際の音のような低レベルの概念よりもゆっくりと変化する。

#### 5) 予測により、新しい入力が予期されたものか予期しないものかが分かる

各 HTM リージョンは新しい事柄の検出器である。各リージョンは次に何が起こるかを予測するので、予期せぬ事が起こったということを知ることができる。HTM は次の入力として起こりうるものを、一つだけではなく一度に多数予測することができる。そのため次に何が起こるかを正確に予測できるわけではないが、もし次の入力がどの予測にも一致しないとき、何か普通でないことが起こったことが分かる。

#### 6) 予測はシステムをノイズにより強くすることができる

HTM が次に何が起こるかを予測したとき、推論をその予測に従う方向へ向かわせる。例えば HTM が話し言葉を処理しているとき、次にどんな音が、単語が、考えが聞こえてくるかを予測する。その予測により、欠落したデータを埋め合わせられる。もしあいまいな音が届いたら、HTM は予測していることに基づいてその音を解釈することで、ノイズが含まれているときの推論を助けることができる。

HTM リージョンの、シーケンス記憶、推論、予測は緊密に統合されている。これらはリージョンの中核機能である。

## 行動<sup>20</sup>

我々の行動は我々が感じることに影響を及ぼす。目を動かすに従って、網膜は変化する入力を受け取る。手や指を動かせば、触った感触が変化する様子が脳に届く。我々のほとんど全ての動作は、我々が感じることを変化させる。センサ入力と筋肉運動は緊密に絡み合っている。

数十年来の主要な見方では、新皮質の単一のリージョンである第1運動野が、新皮質内で運動を指令する場所と考えられてきた。その後、新皮質内のほとんどないしすべてのリージョンは、低レベルの感覚野でさえ、運動に関する出力を出していることが分かった。すべての皮質性リージョンは感覚と運動機能とを統合していると考えられる。

運動指令を生成することは予測することと似ているので、我々は現存のフレームワーク中の各 HTM リージョンに運動出力を加えることができると予期している。しかしながら、これまでのすべての HTM 実装は純粋にセンサ入力に対処するものであって、運動に関する構成要素は含まれていない。

## HTM の実装に向けての進捗状況

我々は HTM の理論的なフレームワークを実用的な技術へと大きく進化させた。我々は HTM 大脳皮質性学習アルゴリズムのいくつかのバージョンを実装・テストし、基本的なアーキテクチャは健全であると分かった。新しいデータでアルゴリズムをテストするに従い、我々はアルゴリズムをより改善し、不足している部分を補ってゆくだらう。これに伴い、この文書も更新する。これに続く3つの章は現在のアルゴリズムの状況を述べる。

この理論の多くの構成要素はまだ実装されていない。注意<sup>21</sup>、リージョン間のフィードバック、特定のタイミング、センサ入力と行動の統合などがある。これらの欠落した構成要素は既に作成されたフレームワークの中にうまく統合されなければならない。

---

<sup>20</sup> behavior

<sup>21</sup> attention

## 第2章： HTM 大脳皮質性学習アルゴリズム

---

本章では HTM リージョンの現状の学習アルゴリズムを説明する。第3章と第4章は疑似コードを用いて学習アルゴリズムの実装方法を説明し、本章では概念的に説明する。

### 用語説明

始める前に、用語を説明すると理解の助けとなるだろう。HTM 学習アルゴリズムを説明するとき、我々はニューロサイエンスの用語を使用する。セル<sup>22</sup>、シナプス<sup>23</sup>、シナプス候補<sup>24</sup>、樹状突起セグメント<sup>25</sup>、カラム<sup>26</sup>などの用語を使用する。学習アルゴリズムは理論上の必要に応じてニューロサイエンスの細部に照らして導かれたので、これらの用語を使うことは理にかなっている。しかしながらアルゴリズムを実装する課程で、我々は性能の問題に直面し、その働きを理解したと感じたときには処理速度を向上させる別の方法を探し求めた。これはときには、生物学的な詳細に厳格にこだわるのではなく、同じ結果が得られさえすれば、これを逸脱することとなった。もし読者がニューロサイエンスの初学者であればこのことは問題とならないであろう。しかしニューロサイエンスの用語に慣れ親しんだ読者であれば、我々が使用する用語が読者の予期するものとはしばしば違うために混乱することもあるだろう。付録の生物学に関する議論では、HTM 学習アルゴリズムとニューロ生物学的に等価なものについて、相違点・類似点を詳細に述べる。ここでは混乱の元となりうる用語の違いのうち、主なものについて説明する。

### セル状態

HTM セルは3つの出力状態を持つ。フィード・フォワード入力によりアクティブな状態、横方向の入力によりアクティブな状態（これは予測を表す）、アクティブでない状態である。最初の出力状態はニューロンのアクション状態による短時間のはげしい出力<sup>27</sup>に相当する。2つ目の出力状態はもっとゆっくりとした、安定した出力<sup>28</sup>に相当

---

<sup>22</sup> cell

<sup>23</sup> synapse

<sup>24</sup> potential synapse

<sup>25</sup> dendrite segment

<sup>26</sup> column

<sup>27</sup> a short burst of action potentials in a neuron

する。我々はこれら2つのアクティブ状態以上の詳細なモデル化の必要性はないと考えた。即ち、個々のアクション状態の強さの程度、アクティビティの発生頻度を表すスカラー量などはモデル化する必要性を見いだせない。分散表現の利用は、セルのアクティビティの程度を表すスカラー量をモデル化することを凌駕するよう感じられる。

## 樹状突起セグメント

HTM セルは比較的リアルな（従ってまた複雑な）樹状突起モデルを持つ。各 HTM セルは理論的に一つの主要樹状突起セグメント<sup>29</sup>と 10~20 個の末梢樹状突起セグメント<sup>30</sup>を持つ。主要樹状突起セグメントはフィード・フォワード入力を受け取り、末梢樹状突起セグメントは周辺のセルからの横方向の入力を受け取る。抑制セルは同じカラム中の全てのセルが類似のフィード・フォワード入力に対して応答するように強制する。単純化のため、各セルごとの主要樹状突起セグメントを取り除き、同じカラム中のすべてのセルで共有する一つの主要樹状突起セグメントで置き換えた。空間プーリング関数（後述）はカラム単位で、共有の樹状突起セグメントに対して作用する。時間プーリング関数はカラム中の個々のセル単位で、末梢樹状突起セグメントに対して作用する。生物学的にはカラムに接続するような樹状突起セグメントは存在しないものの、この単純化は同等の機能をもたらす。

## シナプス

HTM のシナプスは二値のウェイトを持つ。生物学上のシナプスは可変値のウェイトを持つが、確率的・不安定であり、生体ニューロンはシナプスのウェイト値の正確な値に依存しているはずがないということを暗示している。HTM の分散表現及び我々の樹状突起の計算モデルを利用すれば、HTM シナプスに二値のウェイトを割り当てても何ら悪影響はないはずである。シナプスの形成及び切断をモデル化するため、読者には馴染みがないと思われる2つの追加の概念をニューロサイエンスから援用した。一つ目は、シナプス候補の概念である。これは樹状突起セグメントに十分近い位置を通るすべての軸索を表し、シナプスを形成する可能性があるものである。二つ目は、永続値である。これは各シナプス候補に割り当てられたスカラー値である。シナプスの永続値は軸索と樹状突起の間の接続の度合いを表す。その度合は生物学的には、完全に分離した状態から、接続はしていないがシナプスを形成し始めた状態、最小限にシナプスが接続した状態、大きく完全にシナプスが接続された状態に至るまでの範囲を取る。シナプスの

---

<sup>28</sup> a slower, steady rate of action potentials in a neuron

<sup>29</sup> proximal dendrite segment. 樹状突起のうち、ニューロンの中心部に近い部分。

<sup>30</sup> distal dendrite segment. 樹状突起のうち、末端に近い部分。distal は末梢（まっしょう）・末端の意味。ちなみに末梢神経は peripheral nerve という。

永続値は 0.0 から 1.0 までの範囲のスカラー値である。学習にはシナプスの永続値の増加や減少が含まれる。シナプスの永続値がしきい値を超えたら、ウェイト値 1 で接続されたことを表す。しきい値より下回っていたら、ウェイト値 0 で切断されたことを表す。

## 概要

仮に読者が HTM リージョンだったと想像してみよう。貴方の入力数千ないし数万のビットからなる。これらの入力ビットはセンサ入力データや、階層構造の下位の他のリージョンから来たデータである。それらは複雑にオン・オフしている。これらの入力に対して貴方は何が出来るか？

我々はその答えを単純な形態で既に説明した。各 HTM リージョンはその入力の共通のパターンを探し、それらのパターンのシーケンスを学習する。シーケンスの記憶から、各リージョンは予測をする。この高レベルの説明は容易に分かるが、実現にはもう少し説明が必要である。以下の 3 ステップにもう少し詳細化してみよう。

- 1) 入力の疎分散表現を作成する
- 2) 以前の入力の文脈に基づいて、入力の表現を作成する
- 3) 以前の入力の文脈に基づいて、現在の入力からの予測をする

これらのステップについてより詳細に見ていこう。

### 1) 入力の疎分散表現を作成する

リージョンへの入力を想像するには、それを巨大なビット列と考えるとよい。脳内ではこれらはニューロンからの軸索にあたる。任意の時点で、これらの入力のある部分はアクティブ（値 1）、他の部分は非アクティブ（値 0）である。アクティブな入力ビットの比率は変化する。例えば 0% から 60% としよう。HTM リージョンで行う最初の事は、この入力を疎な新しい表現に変換することである。例えば、入力のうち 40% がオンかも知れないが、新しい表現では 2% だけがオンになる。HTM リージョンは論理的にはカラムの集合からなる。各カラムは 1 又はそれ以上のセルから成る。カラムは論理的には 2D の配列状に配置できるが、これは要件ではない。リージョンの各カラムは入力ビットのユニークな部分集合（普通は他のカラムと重なるが、完全に同じ部分集合になることはない）に接続される。結果として、異なる入力パターンからは、レベル全体では異なるカラムのアクティベーションを得る。最も強いアクティベーションを得たカラムは、弱いアクティベーションを得たカラムを抑制、ないし非アクティブ化する。（抑制は非常に局所的範囲からリージョン全体までの範囲で変化する円の円内で起こる）入力

の疎表現は、抑制の後でどのカラムがアクティブでどれが非アクティブであるかによって表される。例えば入力ビットのうちアクティブなビットの数が大幅に変化した場合であっても、相対的に一定の割合のカラムがアクティブになるように抑制関数が定義される。

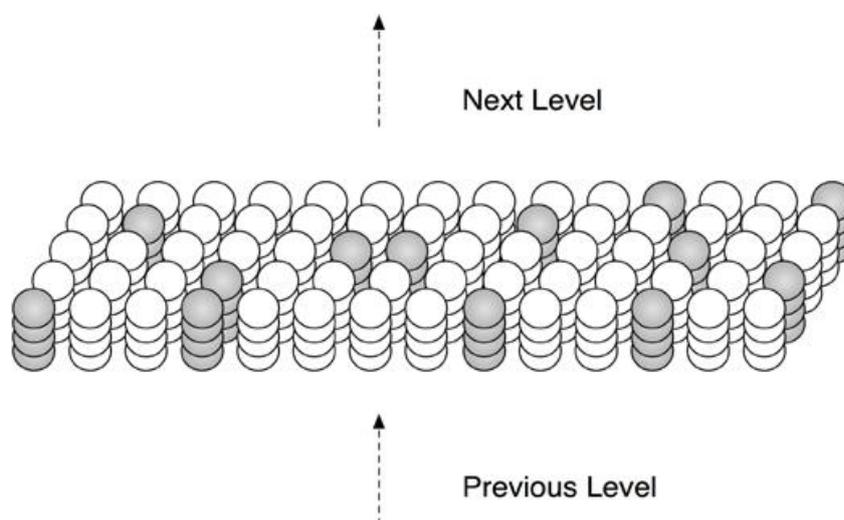


図 2-1 カラムとセルからなる HTM リージョンの例。リージョンの一部のみ表示している。各カラムは入力のユニークな部分集合によるアクティベーションを受け取る。最も強いアクティベーションを受けたカラムが他の弱いアクティベーションを抑制する。結果は入力の疎分散表現である。(アクティブなカラムは灰色で示した)

いま、入力パターンが変化すると想像してみよう。もしほんの少しの入力ビットが変化したなら、いくつかのカラムでは少し多く又は少し少ない入力ビットがオン状態になるが、アクティブなカラムの集合はあまり大幅に変化しないだろう。よって似た入力パターン(アクティブなビットの共通部分が非常に多いもの)からはアクティブなカラムの比較的安定した集合に対応付けられる。コード化がどのくらい安定しているかは、各カラムがどの入力に接続しているかに大きく依存する。この接続関係は、後述する方法で学習する。これらのすべてのステップ(入力の部分集合から各カラムへの接続関係を学習し、各カラムへの入力レベルを決定し、アクティブなカラムの疎な集合をえらぶために抑制すること)を空間プーリングと呼ぶ。この用語は空間的に類似(アクティブなビットの共通部分が多い)のパターンがプーリングされる(それらが共通の表現に互いにグループ化される)ことを意味する。

## 2) 以前の入力の文脈に基づいて、入力の表現を作成する

リージョンで行われる次の機能は、入力をカラムで表現したものを、過去からの状態ないし文脈を含む新しい表現に変換することである。新しい表現は各カラムの一部のセル、普通は1カラムにつき1つのセルをアクティブにすることで得られる(図 2-2)。

「I ate a pear」と「I have eight pears」<sup>31</sup>の二つの話し言葉を聞く場合を考えてみよう。「ate」と「eight」は同音異義語であり、発音が同じである。声に出したときの「ate」と「eight」に対して同じ反応をするニューロンが脳内のどこかにあると考えられる。これら全体で同じ音が耳に入る。また一方、これらの入力が異なり、異なる文脈にあると反応するニューロンが脳内の他のどこかにあると考えられる。「ate」という音に対する表現は「I ate」を聞いたときと「I have eight」を聞いた時とは異なるだろう。「I ate a pear」と「I have eight pears」の二つの文を記憶したと想像してみよう。「I ate...」と聞いたときと「I have eight...」と聞いたときとは、異なる予測をするだろう。よって「I ate...」と聞いたときと「I have eight...」と聞いたときとは異なる内部表現が存在するはずである。

ある入力を異なる文脈では異なるコード変換をするというこの原理は、認知とふるまいの普遍的な特徴であり、HTM リージョンの最も重要な機能の一つである。この能力の重要性はどんなに強調しても、強調し過ぎることはない。

HTM リージョンの各カラムは複数のセルからなっている。同じカラムのすべてのセルは同じフィード・フォワード入力を受け取る。カラム内の各セルはアクティブ又は非アクティブである。アクティブな各カラムごとに、どのセルをアクティブなセルとして選択するかによって、完全に同じ入力に対して異なる文脈では異なる表現をすることができる。例で説明すると分かりやすいだろう。各カラムは4つのセルからなり、各入力は100個のアクティブなカラムで表現されるとしよう。カラムの中で一つのセルだけが一度にアクティブであるとすると、完全に同じ入力に対して $4^{100}$ 通りの表現が可能である。同じ入力は常に同じ組み合わせの100個のカラムがアクティブになるが、文脈が異なればカラム中の異なるセルがアクティブになる。これで同じ入力に対して非常に大きな数の文脈を表現可能となったが、これらの異なる表現はどのくらいユニークであろうか？ $4^{100}$ 個の可能なパターンのうちからランダムに選択した2個は、ほとんどの場合、約25個のセルが重複している。よって同じ入力を異なる文脈で表した2つの表現は、約25個のセルが共通で75個のセルが異なっており、容易に区別することができる。

HTM リージョンの一般的な規則は次のようになる。もしあるカラムがアクティブになると、そのカラム中のすべてのセルを見る。もしそのカラム中の一つ又はそれ以上のセルが既に予測状態であれば、それらのセルだけがアクティブになる。もしそのカラム

<sup>31</sup> 「私は梨を食べる」と「私は8個の梨を持っている」

中のすべてのセルが予測状態でないならば、すべてのセルがアクティブになる。このように考えることができる：ある入力パターンが期待されるなら、システムは予測状態のセルだけをアクティブにすることで期待通りであることを確認する。その入力パターンが期待と違うなら、システムはカラム中のすべてのセルをアクティブにすることで、「予期しない入力が発生したのであらゆる解釈が有りうる」ということを表す。

もし以前の状態が何もないなら、従って文脈も予測もないなら、カラムがアクティブになるときは各カラム内のすべてのセルがアクティブになる。この筋書きは歌の最初の音を聞いたときと似ている。文脈がなければ、普通は次に何が起るかを予測できない：すべての選択肢が有効である。以前の状態があるが入力が予期したものと合致しないときは、アクティブなカラムのすべてのセルがアクティブになる。この決定はカラムごとに行われるので、予測が適合するかしないかは「オール・オア・ナッシング」<sup>32</sup>ではない。

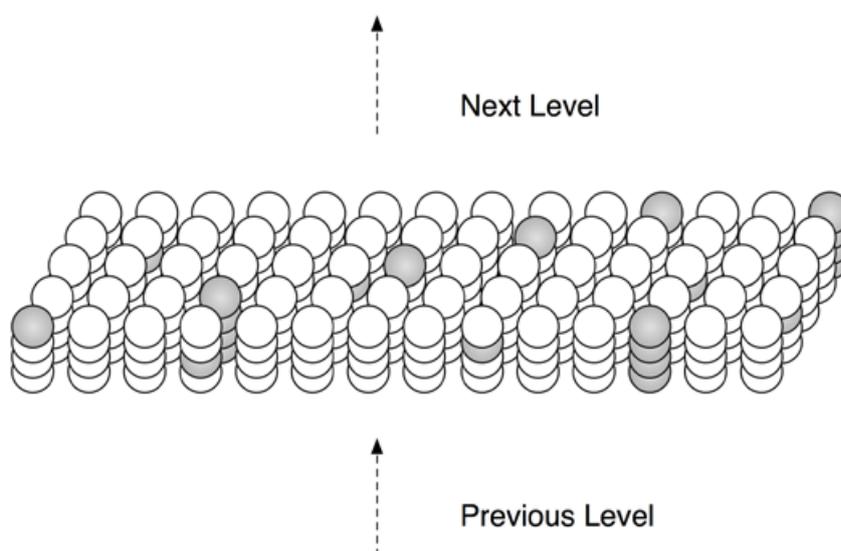


図 2-2 カラムの一部のセルをアクティブにすることで、HTM リージョンは同じ入力の多くの異なる文脈を表現することができる。カラムは予測状態のセルだけをアクティブにする。予測状態のセルがないカラムでは、カラム中のすべてのセルをアクティブにする。図は、あるカラムでは一つのセルだけがアクティブであり、あるカラムではすべてのセルがアクティブである様子を示している。

前述の用語の説明で述べたように、HTM セルは3つの状態を取る。セルがフィード・フォワード入力によってアクティブになったときは単に「アクティブ」の用語で呼

<sup>32</sup> all-or-nothing

ぶ。セルが他の周囲のセルとの横方向の接続によってアクティブになったときは「予測状態」と呼ぶ（図 2-3）。

### 3) 以前の入力の文脈に基づいて、現在の入力からの予測をする

リージョンの最後のステップは次に起こると考えられることを予測することである。予測はステップ 2) で作成した、すべての以前の入力からの文脈を含む表現に基づいて行われる。

リージョンが予測をするときは、将来のフィード・フォワード入力によってアクティブになると考えられるすべてのセルをアクティブ（予測状態）にする。リージョンの表現は疎であるので、同時に複数の予測がなされ得る。例えばカラムのうちの 2% が入力によってアクティブになるとすると、カラムの 20% が予測状態のセルとなることで 10 個の異なる予測がなされ得る。あるいはカラムの 40% が予測状態のセルとなることで、20 個の異なる予測がなされ得る。各カラムが 4 個のセルからなり、一度に一つだけがアクティブになるとすれば、セル全体の 10% が予測状態になることになる。

今後、疎分散表現の章が追加されれば、異なる予測が混じり合っても、リージョンは特定の入力予測されたのかそうでないのかを高い確信さでもって知ることが示されるだろう。

リージョンはどうやって予測をするのか？ 入力パターンが時間と共に変化するとき、カラムとセルの異なる組み合わせが順次アクティブになる。あるセルがアクティブになると、周囲のセルのうちすぐ直前にアクティブだったセルの部分集合への接続を形成する。これらの接続は、そのアプリケーションで必要とされる学習速度に応じて早く形成されたりゆっくりと形成されたりするように調整できる。その後、すべてのセルはこれらの接続を見て、どのセルが同時にアクティブになるかを探さなくてはならない。もし接続がアクティブになったら、セルはそれ自身が間もなくアクティブになることを予測することができ、予測状態に入る。よってある組み合わせのセルがフィード・フォワード入力によってアクティブになると、ひき続いて起こると考えられる他の組み合わせのセルが予測状態になる。このことは、貴方が歌を聞いていて次の音を予測しようとしている瞬間と同様と考えられる。

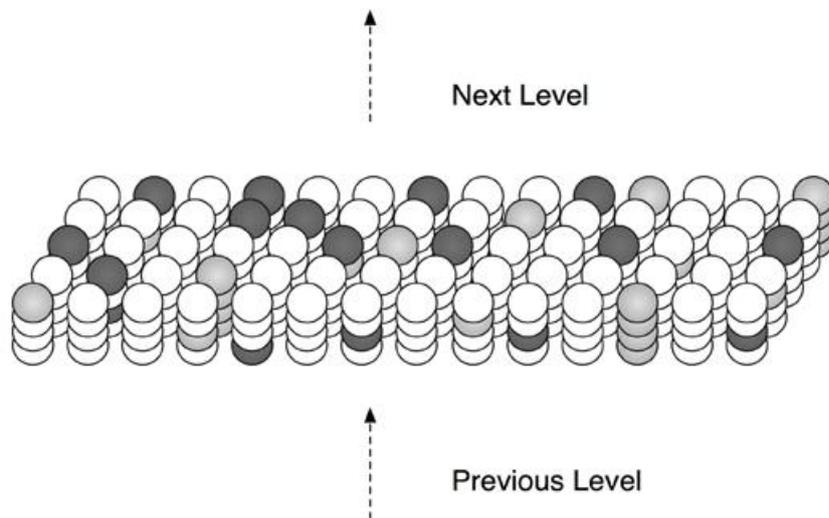


図 2-3 任意の時点で、HTM リージョンのいくつかのセルがフィード・フォワード入力によってアクティブになる（薄い灰色で示した）。他のあるセルは、アクティブなセルからの横方向の入力を受け取って予測状態になる（濃い灰色で示した）。

まとめると、新しい入力が到着すると、アクティブなカラムの疎な部分集合が選択される。各カラムの一つ又はそれ以上のセルがアクティブになり、これはまた同じリージョン内のセル間の接続の学習内容に応じて他のセルを予測状態にする。リージョン内の接続によってアクティブになったセルは、次に何が起こると考えられるかについての予測を表す。次のフィード・フォワード入力が到着すると、他のアクティブなカラムの疎な組み合わせが選択される。新たにアクティブになったカラムが予期したものでないとき、つまりどのセルもそれがアクティブになることを予測しなかったとき、カラム中のすべてのセルをアクティブにする。新たにアクティブになったカラムが一つ又はそれ以上の予測状態のセルを持つなら、それらのセルだけがアクティブになる。リージョンの出力はリージョン内のすべてのセルのアクティブ状態であり、フィード・フォワード入力によってアクティブになったセルと、予測状態のためアクティブになったセルとからなる。

既に述べたように、予測は次の時刻ステップだけではない。HTM リージョンの予測は将来のいくつかのステップに及ぶこともありうる。メロディに例えると、HTM リージョンはメロディの次の音を予測するだけでなく、例えば次の4つの音を予測することも知れない。これは望ましい特徴と言える。リージョンの出力（リージョン内のアクティブ状態のセルと予測状態のセルの和集合）は入力よりもゆっくりと変化する。リージョンがメロディの次の4つの音を予測しているときを想像してみよう。メロディを文字 A, B, C, D, E, F, G のシーケンスで表現する。最初の2音を聞いた後、リージョン

はシーケンスを理解し、予測をし始める。それは C, D, E, F を予測する。B のセル<sup>33</sup>は既にアクティブであるから B, C, D, E, F がそれぞれ 2 つのアクティブな状態のどちらかになる。さらにリージョンが次の音 C を聞く。アクティブ状態のセルと予測状態のセルの集合は C, D, E, F, G を表す。入力パターンは B から C へとまったく違うものに変化した、20%のセルだけが変化している。

HTM リージョンの出力はリージョン内のすべてのセルのアクティブ状態を示すベクトルで表されるので、この例の出力では入力に比べて 5 倍安定している。階層構造に配置されたリージョンでは、階層構造を上に登るに従って時間的な安定性が増加する様子が見られるはずである。

表現に文脈を追加して予測を行う 2 つのステップを「時間プーリング」という用語で表す。パターンのシーケンスに対してゆっくりと変化する出力を生成することで、時間と共に順に現れる異なるパターンを「プールする」<sup>34</sup>。

では、さらに別のレベルで詳細化を進めよう。まずは空間プーリングと時間プーリングで共通の概念から始める。そして空間プーリングに固有の概念と詳細、時間プーリングに固有の概念と詳細の順に説明する。

## 共通概念

空間プーリングの学習と時間プーリングの学習は似ている。どちらの場合の学習も、セル間の接続関係、あるいはシナプスの形成を含む。時間プーリングは同じリージョン内のセル間の接続を学習する。空間プーリングは入力ビットとカラムとのフィード・フォワード接続を学習する。

## 二値ウェイト

HTM のシナプスは 0 又は 1 の効果だけを持つ。多くの他のニューラルネットワークモデルでは 0 から 1 の範囲で変化するスカラー値のウェイトを用いるのと異なり、ウェイトは二値である。

<sup>33</sup> 英文では **cells** と複数形なので、**B** を表すセルは一つではないことが分かる。**A, B, C,...**のそれぞれに対応する疎分散表現は、リージョン全体の 2%のセルの組み合わせで表される。

<sup>34</sup> 何かを貯めこんで蓄積するというニュアンスから、時間に関する情報を蓄積するものという意味と思われる。

## 永続値

シナプスは学習を通じて継続的に形成されあるいは切断される。既に述べたように、各シナプスに（0.0 から 1.0 の）スカラー値を割り当て、接続がどのくらい永続的であるかを表す。接続が強化されれば、永続値は増加する。他の状況では、永続値は減少する。永続値がしきい値（例えば 0.2）を上回れば、シナプスは形成されたと考えられる。永続値がしきい値を下回れば、シナプスは無効である。

## 樹状突起セグメント

シナプスは樹状突起セグメントに接続される。樹状突起には主要と末梢の 2 種類がある。

- 主要樹状突起セグメントはフィード・フォワード入力との間のシナプスを形成する。このタイプのセグメントのアクティブなシナプスは線形に加算され、これによりカラムがフィード・フォワード入力によるアクティブ状態になるか否かが決定される。
- 末梢樹状突起セグメントは同じリージョン内のセル間のシナプスを形成する。各セルはいくつかの末梢樹状突起セグメントを持つ。ある末梢樹状突起セグメント上のアクティブなシナプスの合計がしきい値を超えたら、接続されたセルは予測状態によりアクティブになる。一つのセルに複数の末梢樹状突起セグメントがあるので、セルの予測状態はそれぞれをしきい値で判定した結果の論理和で求められる。

## シナプス候補

既に述べたように、樹状突起セグメントはシナプス候補のリストを持つ。すべてのシナプス候補は永続値を持ち、永続値がしきい値を超えたら有効に機能するシナプスとなる。

## 学習

学習では樹状突起セグメント上のシナプス候補の永続値を増加・減少させる。シナプスの永続性を増加・減少させるために用いられる規則は「ヘブの学習規則」<sup>35</sup>に似ている。例えば、ある樹状突起セグメントがしきい値以上の入力を受け取ったためにセルがアクティブになったときは、そのセグメント上のシナプスの永続値を修正する。シナプスがアクティブであり、従ってセルがアクティブになることに貢献した場合、その永続値を増加させる。シナプスがアクティブではなく、従って貢献しなかった場合、その

---

<sup>35</sup> Hebbian learning rules。「細胞 A の軸索が細胞 B を発火させるのに十分近くにあり、繰り返しあるいは絶え間なくその発火に参加するとき、いくつかの成長過程あるいは代謝変化が一方あるいは両方の細胞に起こり、細胞 B を発火させる細胞の 1 つとして細胞 A の効率が增加する。」

永続値を減少させる。シナプスの永続値を更新する正確な条件は、空間プーリングと時間プーリングとは異なる。詳細は以下に述べる。

では空間プーリング関数と時間プーリング関数に固有の概念について述べる。

## 空間プーリングの概念

空間プーリングの最も基本的な機能はリージョンへの入力を疎なパターンに変換することである。シーケンスを学習して予測をするときに用いられる仕組みでは疎分散パターンから始めることが必要となるため、この機能は重要である。空間プーリングがどれほどうまく計算及び学習をするのかを決定するいくつかの到達目標がある。

### 1) すべてのカラムを使用する

HTM リージョンは入力のごく共通したパターンに対する表現を学習するための固定数のカラムがある。一つの目的は、全体のカラムの数がいくつあったとしても、すべてのカラムが確かに、何か有用なことの表現を学習するという点にある。決してアクティブにならないようなカラムは必要でない。そうならないために、各カラムがその周囲のカラムと相対的にどのくらい頻繁にアクティブになるかを常に監視する。カラムがアクティブになる相対的な頻度が低すぎるときは、そのカラムが勝者となるカラムの集合に含まれ始めるようになるまで、その入力がアクティブになる基準をブースト<sup>36</sup>する。本質的に、すべてのカラムは周囲のカラムと互いに競合しており、入力パターンに対する表現に加わろうとしている。あるカラムがほとんどアクティブにならないときは、そのカラムはもっと積極的になる。そうすると、他のカラムはその入力を変更させられて少しだけ異なる入力パターンを表現し始める。

### 2) 望ましい密度を維持する

リージョンは入力に対する疎な表現を形成する必要がある。最大の入力を受け取ったカラムは周囲のカラムを抑制する。抑制範囲を決める半径は、そのカラムの受容野<sup>37</sup>のサイズに比例する(従ってまた、小さなサイズからリージョン全体に至る範囲を取る)。抑制半径の範囲内では、多くのアクティブな入力を受け取ったわずかなパーセンテージのカラムだけを「勝者」とする。その他のカラムは無効化される。(抑制「半径」の語感<sup>37</sup>は二次元状に配置されたカラムを暗示しているが、この概念は他のトポロジにも適用できる)

<sup>36</sup> boost. 後押しする、増強するなどの意。後述のアルゴリズムでブースト値という変数が出現するため、そのままブーストと訳した。

<sup>37</sup> receptive field

### 3) 些細なパターンを避ける

すべてのカラムが些細ではない入力パターンを表すことが望まれる。この到達目標は、カラムがアクティブになるために必要な最小のしきい値を設定することで達成できる。例えば、しきい値を 50 とすると、カラムがアクティブになるにはその樹状突起セグメント上のアクティブなシナプスが 50 個以上必要であり、これによりあるレベル以上に複雑なパターンだけが表現されることが保証される。

### 4) 余分な接続関係を避ける

よく注意しないと、あるカラムが巨大な数の有効なシナプスを保持することが起こりうる。すると、あまり関連性のない多くの異なる入力パターンに強く反応するようになる。シナプスの異なる部分集合は異なるパターンに反応するだろう。この問題を避けるため、勝者カラムに現在貢献していないシナプスすべてについて、その永続値を減少させる。貢献していないシナプスに確実に十分なペナルティを与えることで、一つのカラムが表現する入力パターンが限定されることを保証し、それはときには一つだけのこともある。

### 5) 自己調整的な受容野

実物の脳は高い可塑性<sup>38</sup>を示す。新皮質のリージョンは、様々な変化に反応してまったく異なる事柄の表現を学習できる。もし新皮質の一部が損傷したら、損傷した部分が表現していたものを他の部分によって表現するように調整される。もし感覚器官が損傷したり変化したりすると、それに関連付けられていた部分の新皮質は何か他のことを表現するように調整される。システムは自己調整的である。我々の HTM リージョンにも同様の柔軟性を求めたい。あるリージョンに 10,000 個のカラムを割り当てたら、入力を 10,000 個のカラムで最適に表現する方法を学習するべきである。あるリージョンに 20,000 個のカラムを割り当てたら、その数を使う最適な方法を学習するべきである。入力の統計的性質が変化したら、カラムはその新たな事実を最適に表現するように変化すべきである。まとめると、HTM の設計者はリージョンに任意のリソースを割り当てることができて、そのリージョンは利用可能なカラムと入力の統計的性質に基づいて入力を最適に表現するという仕事ができるべきである。一般的な規則は、リージョンのカラムがより多くあれば、各カラムは入力より大きくより詳細なパターンを表現するということである。なお一定の粗さを保つが<sup>39</sup>、カラムは普通、より稀にアクティブになる。

<sup>38</sup> plastic. かそせい。物理的な可塑性とは固体に外力を加えて変形させ、力を取り去ってももとに戻らない性質のこと。脳の可塑性とは経験に応じて神経回路の組み換えや再構成を行う能力のこと。柔軟性、適応性。

<sup>39</sup> 粗さ(sparsity)はアクティブになるカラムの割合。カラムの数が増えても粗さは一定ということ。

これらの非常に望ましい到達目標を達成するために、新しい学習規則は必要ない。アクティブでないカラムをブーストし、粗さを一定に保つために周囲のカラムを抑制し、入力に対するしきい値の最小値を設け、多くのシナプス候補を蓄積・維持し、その貢献度に応じてシナプスを追加・削除することで、全体効果としてカラムは望ましい効果を達成するように動的に設定される。

## 空間プーリングの詳細

空間プーリングが行うすべてについて述べる。

- 1) 固定の数のビットからなる入力から始める。これらのビットはセンサからのデータであったり、階層構造の下位の他のリージョンからであったりする。
- 2) この入力を受取る固定の数のカラムをリージョンに割り当てる。各カラムはそれに連結された樹状突起セグメントを持つ。各樹状突起セグメントは入力ビットの部分集合を表すシナプス候補の集合を持つ。各シナプス候補は永続値を持つ。その永続値に基づいて、いくつかのシナプス候補が有効になる。
- 3) 与えられた任意の入力について、アクティブな入力ビットと接続している有効なシナプスの数を各カラムごとに求める。
- 4) アクティブなシナプスの数にブースト値<sup>40</sup>を乗じる。ブースト値は、そのカラムが周囲のものに比べてどのくらい頻繁にアクティブになったかに基づいて動的に決められる。
- 5) ブースト後に最大のアクティベーションを得たカラムは、抑制半径内の固定のパーセンテージのカラム以外のものを無効化する。抑制半径はそれ自体、入力ビットの広がり具合（又はファン・アウト）から動的に決定される。これでアクティブなカラムの疎な集合が得られた。
- 6) アクティブなカラムのそれぞれについて、すべてのシナプス候補の永続値を調節する。アクティブな入力に割り当てられたシナプスの永続値は増加させる。非アクティブな入力に割り当てられたシナプスの永続値は減少させる。永続値の変更により、いくつかのシナプスが有効になったり無効になったりする。

## 時間プーリングの概念

---

<sup>40</sup> boosting factor

時間プーリングがシーケンスを学習し、予測をすることを思い出して欲しい。基本的な方法は、あるセルがアクティブになったら直前にアクティブであった他のセルとの接続を形成することである。これによりセルは、そのセルの接続を調べることでいつそれがアクティブになるかを予測できるようになる。すべてのセルがこれを行えば、全体としてそれらはシーケンスを記憶してそれを思い出し、次に何が起こると考えられるかを予測できる。パターンのシーケンスを記憶するための集中記憶装置はなく、その代わりに記憶は各セルに分散配置される。記憶が分散しているため、システムはノイズや誤りに強くなる。個々のセルが失敗しても、通常それによる影響はわずかないし全くない。なお、時間プーリングが利用している、疎分散表現の重要な特徴を 2, 3 述べておくことは価値がある。

仮想的に、あるリージョンが全部で 10,000 個あるセルのうち、常に 200 個のセルがアクティブになることで表現を形成しているとしよう（任意の時点で 2%のセルがアクティブ）。200 個のアクティブなセルで表される特定のパターンを記憶・理解するにはどうすればよいだろうか。これを行う単純な方法は、関心がある 200 個のアクティブなセルのリストを作成することである。ちょうど同じ 200 個のセルが再びアクティブになったことが分かれば、そのパターンを理解したことになる。しかしながら、200 個のアクティブなセルのうち 20 個だけのリストを作成して、残りの 180 個を無視したとしたらどうだろうか？ 何が起こるだろうか？ 20 個のセルだけを記憶したら、200 個のセルの異なるパターンにおいてそれら 20 個の部分がちょうどアクティブになるようなパターンが数多く存在して、間違いだらけになると読者は考えるかも知れない。しかしそうはならない。パターンは大きくかつ疎であるため（この例では 10,000 個のうち 200 個のセルがアクティブ）、20 個のアクティブなセルを記憶することで 200 個すべてを記憶することとほとんど同じくらいうまく記憶できる。実際のシステムで間違いが起こる可能性は極めて稀で、必要なメモリ量を非常に節約できる。

HTM リージョンのセルはこの特徴を利用している。各セルの樹状突起セグメントは同じセル内の他のセルへの接続関係の集合を持つ。樹状突起セグメントはある時点でのネットワークの状態を理解する手段とするため、これらの接続を形成している。周囲のアクティブなセルは数百から数千あるかも知れないが、樹状突起セグメントが接続しなければならないのはこのうちの 15 から 20 程度に過ぎない。樹状突起セグメントに接続する 15 個のセルがアクティブと分かれば、その大きなパターンが発生しているとほぼ確信できる。このテクニックを「サブサンプリング」と呼び、HTM アルゴリズム全体を通じて利用している。

各セルは多くの異なる分散パターンに関与し、また多くの異なるシーケンスに関与している。ある特定のセルは数十から数百の時間的遷移に関与しているかも知れない。従って各セルは一つではなく、いくつかの樹状突起セグメントを持つ。理想的にはセルが理解したいアクティビティの各パターンごとに一つの樹状突起セグメントを持つこ

とが望ましい。しかしながら実際には、樹状突起セグメントはいくつかの完全に異なるパターンに関して接続を学習することができ、それでもうまく行く。例えば、一つのセグメントが4つの異なるパターンのそれぞれについて20個の接続を持ち、都合80個の接続を持つとする。そして、これらの接続のうち任意の15個がアクティブなときに樹状突起セグメントがアクティブとなるようにしきい値を設定する。これにより誤りが発生する可能性が生じる。異なるパターンが混在することで、樹状突起セグメントのアクティブな接続に関する15個のしきい値に到達する可能性がある。しかしながら、表現の疎な性質により、このような誤りは非常に起こりにくい。

では、10個から20個の樹状突起セグメントを持つセルと数千個のシナプスがどのようにして数百種類のセルのアクティブ状態を理解するのかを見ていこう。

## 時間プーリングの詳細

時間プーリングで行われるステップを数え上げていく。空間プーリングを終えてフィード・フォワード入力を表現するアクティブなカラムの集合が得られたところから始める。

- 1) それぞれのアクティブなカラムについて、カラムの中のセルで予測状態のものを調べ、アクティブにする。すべてのセルが予測状態でないなら、カラム中のすべてのセルをアクティブにする。結果として得られたアクティブなセルの集合は、以前の入力の文脈の下での入力表現である。
- 2) リージョンのすべてのセルの各樹状突起セグメントについて、アクティブなセルに接続されている接続状態のシナプスの数を数える。もしその数がしきい値を超えていけば、その樹状突起セグメントをアクティブとして印を付ける。アクティブな樹状突起セグメントを持つセルを、それがフィード・フォワード入力によって既にアクティブでない限り、予測状態にする。アクティブな樹状突起を持たず、フィード・フォワード入力によりアクティブになっていないセルは、非アクティブにする。以上により、予測状態のセル全体がそのリージョンの予測となる。
- 3) 樹状突起セグメントがアクティブになったとき、そのセグメント上のすべてのシナプスの永続値を更新する。そのアクティブな樹状突起セグメントのすべてのシナプス候補について、アクティブなセルに接続しているシナプスの永続値を増加させ、非アクティブなセルに接続しているシナプスの永続値を減少させる。シナプスの永続値に対するこれらの変更に一時的と印を付ける。

これはセグメントをアクティブにし、従ってまた予測をするほど既に十分に訓練されたシナプスを更新する。しかしながら、可能であればさらに時間をさかのぼって

予測ができるように常に拡張したい。このため、同じセルの二番目の樹状突起セグメントを取り上げ、訓練する。二番目のセグメントとして、以前の時刻ステップのシナプスの状態に最もマッチするものを一つ選択する。このセグメントに対して、以前の時刻ステップのシステムの状態を用いて、アクティブなセルに接続しているシナプスの永続値を増加させ、非アクティブなセルに接続しているシナプスの永続値を減少させる。シナプスの永続値に対するこれらの変更に一時的と印を付ける。

- 4) あるセルがフィード・フォワード入力によって予測状態からアクティブ状態<sup>41</sup>に変化したときはいつも、そのセルに関連付けられているすべてのシナプス候補の「一時的」の印を削除する。従ってフィード・フォワードによってセルがアクティブ化したことを正しく予測したときだけ、シナプスの永続値を更新する。
- 5) セルがアクティブ状態から非アクティブ状態に変化したとき、このセルのすべてのシナプス候補について一時的な永続値の変更を元に戻す。フィード・フォワードによってセルがアクティブ化したことを間違って予測したときはシナプスの永続値を強化したくないため。

フィード・フォワードによってアクティブになったセルだけを処理するのはリージョンの 内部だけ であって、それ以外では予測はさらなる予測を引き起こすことに注意。しかし（フィード・フォワードと予測の）すべてのアクティブなセルはリージョンの出力となり、階層構造の 次の リージョンへと引き継がれる。

## 一次と可変長<sup>42</sup>のシーケンスと予測

空間プーリングと時間プーリングの議論を終える前に、もう一つ大きな論点がある。すべての読者の興味を引かないかも知れないし、また第3章と第4章を理解する上では必要ないけれども。

一つのカラムに対するセルの数を増やしたときないし減らしたときの効果はどうなるだろうか？ 特に、1カラムに1つのセルしかないときは何が起こるだろうか？

以前用いた例では、カラム当たり4セルのアクティブなカラムが100個の場合、入力の表現は $4^{100}$ 通りの異なるコード化が可能であることを示した。従って、同じ入力が多様な文脈の中で出現しても混乱しないようにできる。例えば、もし入力パターンが単語を表すなら、リージョンは同じ単語が何度も使われる多くの文章を混乱することなく

---

<sup>41</sup> 原文は“inactive to active”となっているが、web上のforumで“predictive state to active state”の間違いだったとの訂正があった。(2010/12/14 Sabutai: title “Cortical Algorithms document: praise and suggestions”)

<sup>42</sup> “first order”と“variable order”。前者は一つだけの長さのシーケンスと予測、後者は任意の長さのシーケンスと予測。

記憶できる。「犬」のような単語が異なる文脈の中でユニークな表現を持つことができる。この能力により **HTM** リージョンは可変長の予測が可能となる。

可変長予測は現在起きていることだけではなく、可変の長さの過去の文脈に基づいて予測する。**HTM** リージョンは可変長の記憶である。

カラム当たり 5 セルに増やすと、任意の特定の入力に対して可能なコード化の数は  $5^{100}$  に増加し、 $4^{100}$  よりずっと大きくなる。しかしこれらの数は両方とも非常に大きく、多くの現実的な問題においてこの容量の増加はあまり役に立たないだろう。

しかしながら、カラム当たりのセルの数をこれより少なくすると、大きな違いが生まれる。

もしカラム当たり 1 セルまでになると、文脈の中で表現する能力を失ってしまう。リージョンへの入力は、以前の活動に関係なく常に同じ予測を引き起こす結果となる。カラム当たり 1 セルの場合、**HTM** リージョンの記憶は一次記憶となり、予測は現在の入力だけに基づいて行われる。

一次予測は脳が解くことのできるある種の問題 — 静的空間推論 — に理想的である。既に述べたように、人は短時間だけ画像を見せられたとき、目が後を追うには短かすぎる時間であってもその物体を理解できる。ものを聞くとき、それが何であれ理解するには常にパターンのシーケンスを聞く必要がある。視覚も普通はそれに似ていて、視覚的イメージの流れを処理する必要がある。しかしある条件下では、一瞬のイメージだけでも理解できることがある。

時間的理解と静的理解とでは、異なる推論メカニズムが必要とされているように思われる。一方は可変長の文脈に基づいてパターンのシーケンスを理解し、予測をする必要がある。他方は時間的文脈を使わずに静的な空間的パターンを理解する必要がある。カラム当たり複数のセルを持つ **HTM** リージョンは時間に基づくシーケンスを理解するのに理想的であり、カラム当たり 1 セルの **HTM** リージョンは空間的パターンを理解するのに理想的である。**Numenta** では、カラム当たり 1 セルのリージョンを視覚問題に適用した実験を数多く実施した。これらの実験の詳細はこの章の範囲を超えるが、重要な概念だけ述べる。

**HTM** リージョンにイメージを入力すると、リージョン内のカラムは画素の共通の空間的配列の表現を学習する。学習するパターンの種類は新皮質の **V1** 野（生物学で広く研究されている新皮質のリージョン）で観察されるものと似ていて、概ね、異なる角度の線と角である。動画像で訓練すると、**HTM** リージョンはこれらの基本的な形の遷移を学習する。例えばある箇所に垂直な線があつて、左又は右に移動した垂直な線がそれに続くということがよくある。よく観察されるパターンの遷移は **HTM** リージョンで記憶される。

もしリージョンへの入力画像が、垂直な線が右に移動するものだったら何が起こるだろうか？ カラム当たり 1 セルしかなかったら、線が次に左又は右に現れること<sup>43</sup>を予測できるだろう。線が過去にどこにあったか知っているという文脈を使うことができないため、それが左ないし右に移動していることを知ることはできない。このようなカラム当たり 1 セルのものは、新皮質の「複雑型細胞」<sup>44</sup>のように振舞うと分かるだろう。そのようなセルの予測出力は、その線が左や右に動いていようがいまいが異なる位置にある視覚的な線に対してアクティブになるだろう。このようなリージョンは異なるイメージを区別する能力を保持する一方で、平行移動や大きさの変化に対して安定性を示すことが我々の観察から分かった。このような振る舞いは、空間的不変性（同じパターンの異なる見方を理解すること）のために必要である。

もし同じ実験をカラム当たり複数のセルを持つ HTM リージョンに対して行えば、そのセルが新皮質の「方位選択性複雑型細胞」<sup>45</sup>のように振舞うと分かるだろう。セルの予測出力は左に移動する線や右に移動する線に対してアクティブになるが、両方に対してはアクティブにならないだろう。

これらをまとめて、我々は次のような仮説を立てた。新皮質は一次と可変長の両方の推論及び予測をしなければならない。新皮質の各リージョンには 4 又は 5 層のセルがある。層は様々な形に異なっているが、それらはすべてカラム単位で応答する性質を共有しており、その層内で水平方向に大きな接続性を持っている。新皮質のセルの層はそれぞれ、この章で述べたような HTM の推論と学習に似たことを実行しているのはいか、と我々は推測した。異なる層のセルは異なる役割を演じている。例えば解剖学によれば第 6 層は階層構造のフィードバックを形成し、第 5 層は運動の動作に関わっている。セルの 2 つの主要なフィード・フォワード層は第 4 層と第 3 層である。第 4 層と第 3 層の一つの違いは、第 4 層のセルが独立に、即ちカラムの中で 1 セルだけが動作するのに対して、第 3 層のセルはカラムの中で複数のセルが動作することだろうと我々は推測した。よってセンサ入力に近い新皮質のリージョンは一次記憶と可変長記憶の両方を持つ。一次シーケンス記憶（だいたい第 4 層のニューロンに対応する）は空間的に不変の表現を形成するのに役立つ。可変長シーケンス記憶（だいたい第 3 層のニューロンに対応する）は動画像の推論と予測に役立つ。

まとめると我々は、この章で述べたようなアルゴリズムは新皮質のニューロンのすべての層で働いているという仮説を立てた。新皮質の層の詳細は大きく違って、フィード・フォワードとフィードバック、注意<sup>46</sup>、運動動作<sup>47</sup>に関する異なる役割を演じ

<sup>43</sup> 「移動する」ことは予測できないが、隣の位置に「出現する」ことは予測できるということ。

<sup>44</sup> complex cell

<sup>45</sup> directionally-tuned complex cell

<sup>46</sup> attention

<sup>47</sup> motor behavior

ている。センサ入力に近いリージョンでは、一次記憶を実行するニューロンの層が空間的不変性に有利であるため役に立つ。

**Numenta** では、一次（カラム当たり 1 セル）の **HTM** リージョンを画像認識問題に適用する実験をした。我々はまた、可変長（カラム当たり複数セル）の **HTM** リージョンに可変長のシーケンスを理解・予測させる実験をした。将来的には、これらを一つのリージョンに混在させ、他の目的にもアルゴリズムを拡張することを試みることは理にかなっている。しかしながら、一つの層と等価なカラム当たり複数セルの構造が、単体であれ複数階層であれ、多くの興味深い問題を取り扱いうると我々は信じている。

## 第3章：空間プーリングの実装と疑似コード

---

本章では空間プーリング関数<sup>48</sup>の最初の実装の疑似コードを詳細に示す。このコードの入力は、センサー・データ又は前のレベルからのバイナリ配列である。このコードは `activeColumns(t)` を計算する。`activeColumns(t)` は時刻  $t$  において、フィード・フォワード入力に対して選択されたカラムのリストである。このリストは、次章で述べる時間プーリング関数の入力として送られる。即ち、`activeColumns(t)` は空間プーリング関数の出力である。

疑似コードは3つのフェーズに明確に分かれる。これらは順に実行される。

フェーズ 1: 各カラムについて、現在の入力のオーバーラップを計算する。

フェーズ 2: 抑制の後に勝者となったカラムを計算する。

フェーズ 3: シナプスの永続値と内部変数を更新する。

空間プーリングの学習はオンライン<sup>49</sup>で行われるが、フェーズ 3 を単にスキップすることで学習をしないようにすることもできる。

以下、3つのフェーズのそれぞれについて疑似コードを示す。疑似コードの中で使用されている様々なデータ構造や補助関数は本章の最後に示す。

### 初期化

最初の入力を受け取る前に、各カラムの最初のシナプス候補のリストを計算してリージョンを初期化する。これは入力配列の中からランダムに選択された入力位置のリストで構成される。各入力シナプスで表現され、ランダムな永続値が割り当てられる。ランダムな永続値は二つの条件を満たすように選ばれる。第一に、その値は `connectedPerm` (シナプスが「接続している」と判定される最小の永続値) の前後の狭い範囲から選ばれる。これにより、訓練を少ない回数繰り返すだけで、シナプス候補が接続 (ないし切断) することを可能とする。第二に、各カラムは入力リージョン上で自然な中心位置があり、永続値はこの中心に向かってバイアスを加えられる。(中心付近ではより高い値を与えられる)

---

<sup>48</sup> `spatial pooler function`

<sup>49</sup> `online`。推論の計算と学習の計算を分離せずに、同時に行うということ。

## フェーズ 1: オーバラップ

最初のフェーズでは、与えられた入力ベクトルについて、そのベクトルと各カラムのオーバラップを計算する。各カラムのオーバラップは、アクティブな入力と接続されたシナプスの数を数え、それにブースト値を掛けたものである。もしこの値が `minOverlap` を下回っている場合、オーバラップは 0 とする。

```
1. for c in columns
2.
3.     overlap(c) = 0
4.     for s in connectedSynapses(c)
5.         overlap(c) = overlap(c) + input(t, s.sourceInput)
6.
7.     if overlap(c) < minOverlap then
8.         overlap(c) = 0
9.     else
10.        overlap(c) = overlap(c) * boost(c)
```

## フェーズ 2: 抑制

第 2 のフェーズでは、抑制の後に勝者となったカラムを計算する。`desiredLocalActivity` は勝者となるカラムの数を制御するパラメータである。例えば、`desiredLocalActivity` を 10 とすると、抑制半径の範囲内においてカラムのオーバラップ値が高い順に 10 位以内のカラムが勝者となる。

```
11. for c in columns
12.
13.     minLocalActivity = kthScore(neighbors(c), desiredLocalActivity)
14.
15.     if overlap(c) > 0 and overlap(c) ≥ minLocalActivity then
16.         activeColumns(t).append(c)
17.
```

## フェーズ 3: 学習

第 3 のフェーズは学習を実行する。これによりすべてのシナプスの永続値は必要に応じて更新され、ブースト値と抑制半径を更新する。

主要な学習規則は 20-26 行目で実装されている。勝者となったカラムのそれぞれについて、もしあるシナプスがアクティブであればその永続値をインクリメントし、その他の場合はデクリメントする。永続値は 0 から 1 の範囲内に束縛される。

28-36 行目ではブーストを実装している。カラムが接続を学習するための二つの独立したブースト機構がある。あるカラムがあまり勝者となっていない (`activeDutyCycle` で観測される) とき、そのブースト値をインクリメントする (30-32 行目)。一方、あるカラムのシナプスがどの入力ともあまりオーバーラップしない<sup>50</sup> (`overlapDutyCycle` で観測される) とき、その永続値がブーストされる (34-36 行目)。ノート：学習モードがオフになると、ブースト値は固定される。

フェーズ 3 の最後に、抑制半径を再計算する (38 行目)。

```
18. for c in activeColumns(t)
19.
20.     for s in potentialSynapses(c)
21.         if active(s) then
22.             s.permanence += permanencInc
23.             s.permanence = min(1.0, s.permanence)
24.         else
25.             s.permanence -= permanenceDec
26.             s.permanence = max(0.0, s.permanence)
27.
28. for c in columns:
29.
30.     minDutyCycle(c) = 0.01 * maxDutyCycle(neighbors(c))
31.     activeDutyCycle(c) = updateActiveDutyCycle(c)
32.     boost(c) = boostFunction(activeDutyCycle(c), minDutyCycle(c))
33.
34.     overlapDutyCycle(c) = updateOverlapDutyCycle(c)
35.     if overlapDutyCycle(c) < minDutyCycle(c) then
36.         increasePermanences(c, 0.1*connectedPerm)
37.
38. inhibitionRadius = averageReceptiveFieldSize()
39.
```

<sup>50</sup> オーバーラップ値が小さい

## データ構造と補助関数

以下の変数とデータ構造が疑似コードで使用されている。

columns	すべてのカラムのリスト
input(t,j)	時刻 $t$ におけるこのレベルへの入力。 $j$ 番目の入力がオンのとき、 $\text{input}(t, j)$ は1である。
overlap(c)	ある入力パターンに対する、カラム $c$ の空間プーリング・オーバーラップ
activeColumns(f)	フィード・フォワード入力により勝者となったカラムの添え字のリスト
desiredLocalActivity	抑制ステップの後に勝者となるカラムの数を制御するパラメータ
inhibitionRadius	カラムに接続された受容野 <sup>51</sup> のサイズの平均値
neighbors(c)	カラム $c$ から $\text{inhibitionRadius}$ の範囲内にあるすべてのカラムのリスト
minOverlap	抑制ステップで処理対象となるべきカラムのアクティブな入力の最小の数 <sup>52</sup>
boost(c)	学習のときに計算される、カラム $c$ のブースト値。この値はアクティブでないカラムのオーバーラップ値を増加させるために使われる。
synapse	シナプスを表すデータ構造。永続値と接続元の入力の添え字からなる。
connectedPerm	もしあるシナプスの永続値がこの値よりも大きければ、接続していると判定される
potentialSynapses(c)	シナプス候補とその永続値のリスト
connectedSynapses(c)	$\text{potentialSynapses}(c)$ の部分集合で、永続値が $\text{connectedPerm}$ 以上のものからなる。これらは現在カラム $c$ に接続されているフィード・フォワード入力である。
permanenceInc	学習時にシナプスの永続値を増加させる増分値
permanenceDec	学習時にシナプスの永続値を減少させる減少値

<sup>51</sup> 巻末の用語の説明参照

<sup>52</sup> あるカラムへのアクティブな入力がこの数以上であれば、抑制ステップで処理対象となる。

<code>activeDutyCycle(c)</code>	抑制の後にカラムc がアクティブになった頻度を表す移動平均値 (例えば直近1000回繰り返した間にアクティブになった回数)
<code>overlapDutyCycle(c)</code>	カラムc がその入力に対して有意なオーバーラップ値 (即ち、 <code>minOverlap</code> より大きな値) になった頻度を表す移動平均値 (例えば直近1000回繰り返した間に有意なオーバーラップ値になった回数)
<code>minDutyCycle(c)</code>	最小限望まれるセルの発火頻度を表す変数。あるセルの発火頻度がこの値を下回れば、それはブーストされる。この値はその付近のカラムの最大の発火頻度の1%として計算する。

以下の補助関数が疑似コードで使用されている。

<code>kthScore(cols, k)</code>	与えられたカラムのリストに対して、k番目のオーバーラップ値を返す
<code>updateActiveDutyCycle(c)</code>	抑制の後にカラムc がアクティブになった頻度の移動平均を計算する
<code>updateOverlapDutyCycle(c)</code>	カラムc のオーバーラップ値が <code>minOverlap</code> より大きくなった頻度の移動平均を計算する
<code>averageReceptiveFieldSize()</code>	カラムに接続された受容野の半径についての、すべてのカラムの平均値。カラムに接続された受容野は接続されたシナプス (永続値 $\geq$ <code>connectedPerm</code> のもの) だけが含まれる。これはカラム間の横方向の抑制範囲を決めるために用いられる。
<code>maxDutyCycle(cols)</code>	与えられたカラムのリストのうち、 <code>activeDutyCycle</code> が最大のものを返す
<code>increasePermanences(c, s)</code>	カラムc のすべてのシナプスの永続値をスケール因子s に従って増加させる
<code>boostFunction(c)</code>	カラムc のブースト値を返す。ブースト値は1以上のスカラ値である。 <code>activeDutyCycle(c)</code> が <code>minDutyCycle(c)</code> より大きければブースト値は1。 <code>activeDutyCycle</code> が <code>minDutyCycle</code> より下回り始めて以降は、ブースト値はリニアに増加する。

## 第4章： 時間プーリングの実装と疑似コード

---

本章では時間プーリング関数<sup>53</sup>の最初の実装の疑似コードを詳細に示す。このコードの入力は空間プーリング関数で計算した `activeColumns(t)` である。このコードは時刻 `t` における各セルのアクティブ状態及び予測状態を計算する。アクティブ状態と予測状態の論理和が時間プーリング関数の出力であり、次のレベルの入力である。

疑似コードは3つのフェーズに明確に分かれる。これらは順に実行される。

フェーズ 1: 各セルについてアクティブ状態 `activeState(t)` を計算する。

フェーズ 2: 各セルについて予測状態 `predictiveState(t)` を計算する。

フェーズ 3: シナプスを更新する。

フェーズ 3 は学習するときだけに必要となる。しかしながら、空間プーリングのときとは異なり、学習が有効のときはフェーズ 1 と 2 も学習特有の操作をいくつか含んでいる。時間プーリングは空間プーリングよりかなり複雑であるため、まずは時間プーリングの推論だけのバージョンを示し、次に推論と学習を含むバージョンを示す。補助関数は疑似コードの後に、本章の最後に示す。

### 時間プーリング疑似コード： 推論だけのバージョン

#### フェーズ 1

第1のフェーズでは、各セルのアクティブ状態を計算する。勝者となった各カラムについて、どのセルをアクティブにするのかを決定する。フィード・フォワード入力がいずれかのセルによって予測された（即ち、前回の時刻ステップで順序セグメントによって `predictiveState` が 1 になった）とき、それらのセルをアクティブにする（4-9 行目）。フィード・フォワード入力予測されなかった（どのセルも `predictiveState` がオンにならなかった）とき、そのカラムのすべてのセルをアクティブにする（11-13 行目）。

---

<sup>53</sup> temporal pooler function

```

1.  for c in activeColumns(t)
2.
3.      buPredicted = false
4.      for i = 0 to cellsPerColumn - 1
5.          if predictiveState(c, i, t-1) == true then
6.              s = getActiveSegment(c, i, t-1, activeState)
7.              if s.sequenceSegment == true then
8.                  buPredicted = true
9.                  activeState(c, i, t) = 1
10.
11.     if buPredicted == false then
12.         for i = 0 to cellsPerColumn - 1
13.             activeState(c, i, t) = 1

```

## フェーズ 2

第2のフェーズでは、各セルの予測状態を計算する。セルのどれかのセグメントがアクティブになると、そのセルの `predictiveState` がオンになる。即ち、十分な数の横方向の接続先が、フィード・フォワード入力によって現在発火していればオンになる。

```

14. for c, i in cells
15.     for s in segments(c, i)
16.         if segmentActive(c, i, s, t) then
17.             predictiveState(c, i, t) = 1

```

## 時間プーリング疑似コード： 推論と学習を含むバージョン

### フェーズ 1

第1のフェーズでは、勝者となったカラム中の各セルの `activeState` を計算する。それらのカラムについて、コードはさらにカラムごとに一つのセルを学習セル (`learnState`)として選択する<sup>54</sup>。そのロジックは以下の通り。フィード・フォワード入力がいずれかのセルによって予測された（即ち、順序セグメントによって `predictiveState`

<sup>54</sup> `lcChosen` は学習セルが選択されたこと (`learn cell chosen`)を表し、`(c, i)` が選択された学習セル、そして `learnState(c, i, t) = 1` に設定されることでこのセルが学習セルとして選択されたことを記憶する。

が 1 になった) とき<sup>55</sup>、それらのセルをアクティブにする (23-27 行目)。そのセグメントが、learnState がオンのセルによってアクティブになった場合、そのセルは学習セルとして選択される (28-30 行目)。フィード・フォワード入力が予測されなかったとき、そのカラムのすべてのセルをアクティブにする (32-34 行目)。さらに、ベストマッチセルが学習セルとして選択され (36-41 行目)、新しいセグメントがそのセルに追加される。

```
18. for c in activeColumns(t)
19.
20.     buPredicted = false
21.     lcChosen = false
22.     for i = 0 to cellsPerColumn - 1
23.         if predictiveState(c, i, t-1) == true then
24.             s = getActiveSegment(c, i, t-1, activeState)
25.             if s.sequenceSegment == true then
26.                 buPredicted = true
27.                 activeState(c, i, t) = 1
28.                 if segmentActive(s, t-1, learnState) then
29.                     lcChosen = true
30.                     learnState(c, i, t) = 1
31.
32.         if buPredicted == false then
33.             for i = 0 to cellsPerColumn - 1
34.                 activeState(c, i, t) = 1
35.
36.         if lcChosen == false then
37.             l,s = getBestMatchingCell(c, t-1)
38.             learnState(c, i, t) = 1
39.             sUpdate = getSegmentActiveSynapses (c, i, s, t-1, true)
40.             sUpdate.sequenceSegment = true
41.             segmentUpdateList.add(sUpdate)
```

<sup>55</sup> buPredicted はフィード・フォワード入力が予測されたこと (bottom-up predicted) を表す。  
(bottom-up は feed-forward と同義で、「フィード・フォワード」と訳している)

## フェーズ 2

第2のフェーズでは、各セルの予測状態を計算する。セルのどれかのセグメントがアクティブになると、そのセルの `predictiveState` がオンになる。即ち、十分な数の横方向の接続先が、フィード・フォワード入力によって現在アクティブであればオンになる。この場合、そのセルは以下の変更を待ち行列に加える： a) 現在アクティブなセグメントを強化<sup>56</sup>する（47-48 行目）、b) このアクティベーション<sup>57</sup>を予測し得たセグメント（即ち、前回の時刻ステップでアクティビティに（弱いかもしれないが）マッチしたセグメント）を強化する（50-53 行目）。

```
42. for c, i in cells
43.     for s in segments(c, i)
44.         if segmentActive(s, t, activeState) then
45.             predictiveState(c, i, t) = 1
46.
47.             activeUpdate = getSegmentActiveSynapses(c, i, s, t, false)
48.             segmentUpdateList.add(activeUpdate)
49.
50.             predSegment = getBestMatchingSegment(c, i, t-1)
51.             predUpdate = getSegmentActiveSynapses(
52.                 c, i, predSegment, t-1, true)
53.             segmentUpdateList.add(predUpdate)
```

## フェーズ 3

第3の、そして最後のフェーズでは、実際に学習を実施する。待ち行列に追加されたセグメントの更新は、フィード・フォワード入力を得てセルが学習セルとして選択されたときに実施される（55-57 行目）。そうではなく、もしセルが何らかの理由で予測を停止した場合、そのセグメントをネガティブ<sup>58</sup>に強化する（58-60 行目）。

<sup>56</sup> reinforcement。フェーズ 3 で強化学習をする処理対象を `segmentUpdateList` に記憶する。

<sup>57</sup> 44 行目の if 文の条件が成立したこと

<sup>58</sup> 逆に弱める方向に学習する

```
54. for c, i in cells
55.     if learnState(s, i, t) == 1 then
56.         adaptSegments (segmentUpdateList(c, i), true)
57.         segmentUpdateList(c, i).delete()
58.     else if predictiveState(c, i, t) == 0 and predictiveState(c, i, t-1)==1 then
59.         adaptSegments (segmentUpdateList(c, i), false)
60.         segmentUpdateList(c, i).delete()
61.
```

## 実装の詳細と用語説明

この節では時間プーリングの実装と用語の詳細を述べる。各セルは二つの数値でアクセスする。カラムの添字  $c$  と、セルの添字  $i$  である。セルは樹状突起セグメントのリストを保持する。各セグメントはシナプスのリストと、各シナプスごとに永続値を保持する。セルのシナプスに対する変更は、セルがフィード・フォワード入力によってアクティブになるまでは一時的とマークされる。これらの一時的な変更は `segmentUpdateList` によって保持される。各セグメントはまた、論理値のフラグ `sequenceSegment` を保持する。これはそのセグメントが次の時刻ステップにおけるフィード・フォワード入力を予測するかどうかを示している。

シナプス候補の実装は空間プーリングの実装とは異なっている。空間プーリングでは、シナプス候補の完全なリストが明示的に示される。時間プーリングでは各セグメントが固有のシナプス候補の（ときには大きな）リストを保持することができる。実際には各セグメントごとに大きなリストを管理することは、計算量が大きくメモリ消費が集中する。そこで我々は、時間プーリングでは学習の際に各セグメントごとにアクティブなシナプスをランダムに追加する（`newSynapseCount` パラメータで制御する）。この最適化はすべてのシナプス候補のリストを維持管理するのと同様の効果があり、しかも新たな時間的パターンを学習できる可能性を維持しながらもセグメント毎のリストはずっと小さくなる。

疑似コードはまた、異なる時刻ステップのセル状態の推移を追うために小さな状態遷移マシンを使用する。各セルごとに三つの異なる状態を維持管理する。配列 `activeState` と `predictiveState` は各セルの各時刻ステップごとのアクティブ状態及び予測状態の推移を追う。配列 `learnState` はどのセルの出力が学習のときに使用されるかを決定する。入力が予測されなかったときは、その特定のカラムのすべてのセルが同じ時刻ステップ内に同時にアクティブになる。これらのセルのうちの一つだけ（入力に最もマッチするセル）で `learnState` がオンになる。`learnState` がオンのセルだけにつ

いて、シナプスを追加する（これは樹状突起セグメントの中で完全にアクティブになったカラムを強調しすぎないようにするためである）。

以下のデータ構造が時間プーリング疑似コードで使用されている。

<b>cell(c,i)</b>	すべてのセルのリスト。 <b>i</b> と <b>c</b> で指し示される。
<b>cellsPerColumn</b>	各カラムのセルの数
<b>activeColumns(f)</b>	フィード・フォワード入力によって勝者となったカラム（空間プーリング関数の出力）のインデックスのリスト
<b>activeState(c, i, t)</b>	各セルが一つずつ持つ論理値のベクトル。カラム <b>c</b> セル <i>i</i> 時刻 <b>t</b> におけるアクティブ状態を表す。これは現在のフィード・フォワード入力と過去の時間的文脈から与えられる。 <b>activeState(c, i, t)</b> が1なら、そのセルは現在フィード・フォワード入力を持ち、適切な時間的文脈を持つ。
<b>predictiveState(c, i, t)</b>	各セルが一つずつ持つ論理値のベクトル。カラム <b>c</b> セル <i>i</i> 時刻 <b>t</b> における予測状態を表す。これは他のカラムのフィード・フォワード状態と過去の時間的文脈から与えられる。 <b>predictiveState(c, i, t)</b> が1なら、そのセルは現在の時間的文脈からフィード・フォワード入力を予測している。
<b>learnState(c, i, t)</b>	カラム <b>c</b> セル <i>i</i> が学習対象のセルとして選択されたことを表す論理値
<b>activationThreshold</b>	あるセグメントをアクティブにするしきい値。あるセグメントに接続されたアクティブなシナプスの数が <b>activationThreshold</b> より大きければ、セグメントはアクティブになる。
<b>learningRadius</b>	横方向の接続を持つ、時間プーリングセルの周囲の領域の範囲
<b>initialPerm</b>	シナプスの永続値の初期値
<b>connectedPerm</b>	あるシナプスの永続値がこの値より大きければ、それが接続していることを表す
<b>minThreshold</b>	学習の際の、アクティブなセグメントの最小数
<b>newSynapseCount</b>	学習のときにセグメントに追加されるシナプスの最大数
<b>permanenceInc</b>	アクティビティによる学習が発生したとき、シナプスの永続値を増加させる量
<b>permanenceDec</b>	アクティビティによる学習が発生したとき、シナプスの永続値を減少させる量

<b>segmentUpdate</b>	与えられたセグメントを更新するときに必要な情報を保持するデータ構造で、以下の3項目からなる。a) セグメントの添字（新しいセグメントのときは-1）、b) 既存のアクティブなシナプスのリスト、c) このセグメントが順序セグメントとマークされべきかどうかを表すフラグ（デフォルト値はfalse）。
<b>segmentUpdateList</b>	<b>segmentUpdate</b> 構造体のリスト。 <b>segmentUpdateList(c,i)</b> はカラムc セルi の変更内容のリストである。

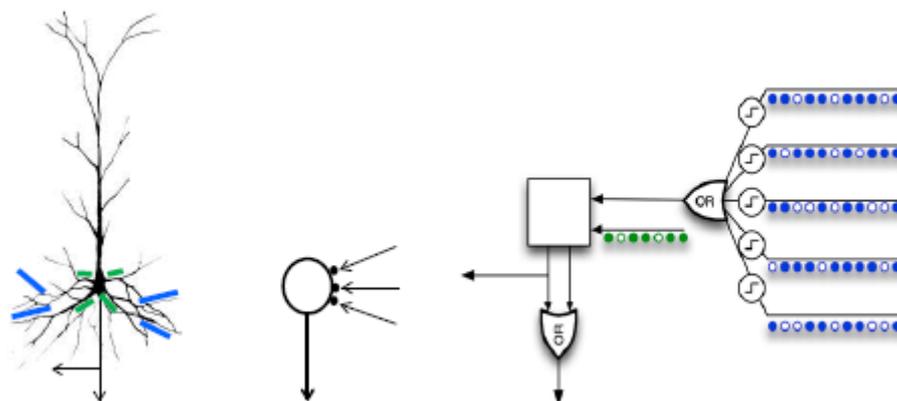
以下の補助関数が上記のコードで使用されている。

<b>segmentActive</b> (s, t, state)	セグメントs 時刻t において、state で与えられた状態によってアクティブになった接続状態のシナプスの数が <b>activationThreshold</b> より大きい時、真を返す。state パラメータは <b>activeState</b> 又は <b>learnState</b> である。
<b>getActiveSegment</b> (c, i, t, state)	与えられたカラムc セルi について、 <b>segmentActive(s,t, state)</b> が真になるセグメントの添字を返す。もし複数のセグメントがアクティブであれば、順序セグメントがあればそれが優先する。そうでなければもっともアクティビティが高いものが優先する。
<b>getBestMatchingSegment</b> (c, i, t)	与えられたカラムc セルi について、アクティブなシナプスが最も多いセグメントを探す。この関数は積極的にベストマッチを見つける。シナプスの永続値は <b>connectedPerm</b> より小さくても許される。アクティブなシナプスの数は <b>activationThreshold</b> より小さくても許される。ただし、 <b>minThreshold</b> より大きくなければならない。この関数はセグメントの添字を返す。もしセグメントが見つからなかったら -1 を返す。
<b>getBestMatchingCell</b> (c)	与えられたカラムについて、ベストマッチセグメント（上記参照）を持つセルを返す。もしどのセルにもマッチするセグメントがなければ、最もセグメントの数が少ないセルを返す。

<p><code>getSegmentActiveSynapses(c, i, t, s, newSynapses=false)</code></p>	<p>セグメント <code>s</code> に対して提示された変更のリストを保持する <code>segmentUpdate</code> 構造体を返す。 <code>activeSynapses</code> を、接続元のセルの時刻 <code>t</code> における <code>activeState</code> 出力=1 であるようなアクティブなシナプスのリストとする (<code>s=-1</code> のときはセグメントが存在しないため、このリストは空である)。 <code>newSynapses</code> はオプション引数でデフォルト値は <code>false</code> である。 <code>newSynapses</code> が真の時は、 <code>newSynapseCount - count(activeSynapses)</code> 個のシナプスが <code>activeSynapses</code> に追加される。これらのシナプスは、時刻 <code>t</code> において <code>learnState</code> 出力=1 であるセルの中からランダムに選択される。</p>
<p><code>adaptSegments(segmentList, positiveReinforcement)</code></p>	<p>この関数は <code>segmentUpdate</code> のリスト要素について繰り返し、各セグメントを強化をする。 <code>segmentUpdate</code> の各要素について、以下の変更が実行される。 <code>positiveReinforcement</code> が真のとき、アクティブなリスト上のシナプスの永続値は <code>permanenceInc</code> だけ増加させる。他のすべてのシナプスは永続値を <code>permanenceDec</code> だけ減少させる。 <code>positiveReinforcement</code> が偽のとき、アクティブなリスト上のシナプスは永続値を <code>permanenceDec</code> だけ減少させる。これらの処理の後、 <code>segmentUpdate</code> の中にまだ存在しないシナプスについて、永続値 <code>initialPerm</code> にて追加する。</p>

## 付録 A: 生体ニューロンと HTM セルの比較

---



上の画像は左側が生体ニューロンの写真、中央が単純な人工ニューロン、右側が HTM のニューロンあるいはセルを示している。この付録の目的は HTM セルを実物のニューロンや単純な人工ニューロンと比べることで、それをより深く理解し、またどのように動作するかを示すことにある。

実物のニューロンは途方もなく複雑で変化に富んでいる。ここではその最も普遍的な原理に注目し、また我々のモデルに関わる部分に限定する。実物のニューロンの多くの詳細は無視するものの、HTM 大脳皮質性学習アルゴリズムで用いられているセルは多くのニューラルネットワークで用いられている人工ニューロンよりもはるかに現実に即している。HTM セルに含まれるすべての要素は HTM リージョンの演算に必要である。

### 生体ニューロン

ニューロンは脳内で情報を伝えるセル<sup>59</sup>である。上記左の画像は標準的な興奮性のニューロンである。ニューロンの外見の大部分は枝分かれした樹状突起で占められている。ニューロンへのすべての興奮性の入力、樹状突起に沿って配置されたシナプスを經由して入る。近年、ニューロンに関する知識は有為な発展を遂げた。最大の発見は、ニューロンの樹状突起が入力を細胞体<sup>60</sup>に繋ぐ単なる導線ではないと分かったことで

---

<sup>59</sup> cell を本書では一貫して「セル」と訳したが、生物の細胞を意味する。

<sup>60</sup> cell body。ニューロン中央の膨らんだ部分。

ある。今では樹状突起はそれ自体、複雑な非線形処理部品であることが分かっている。HTM 大脳皮質性学習アルゴリズムはこの非線形の特徴を利用している。

ニューロンはいくつかの部分に分かれている。

## 細胞体

細胞体はニューロンの中央にある小さな体積を持つ部分である。セルの出力、軸索<sup>61</sup>は、細胞体から出ている。セルへの入力も樹状突起に沿って配置されたシナプスであり、それは細胞体に入力される。

## 主要樹状突起<sup>62</sup>

細胞体に最も近い部分にある樹状突起の枝は主要樹状突起と呼ばれる。図では主要樹状突起のいくつかを緑色の線で示した。

主要樹状突起上の複数のアクティブなシナプスは、細胞体に対して概ね線形の加算的効果がある。5つのアクティブなシナプスは1つのアクティブなシナプスに比べて概ね5倍の脱分極<sup>63</sup>を細胞体に対して引き起こす。対照的に、ある一つのシナプスが後続の素早い活動電位<sup>64</sup>によって繰り返しアクティブになったとしても、2番目、3番目と続く活動電位による細胞体への影響は、最初のものよりずっと小さくなる。

このため、主要樹状突起への入力は細胞体に対して線形に加わることで、単一のシナプスに届いた複数の素早いパルスの影響は一つのパルスより少し大きいだけであることが言える。

新皮質のリージョンへのフィード・フォワード接続は主要樹状突起に優先的に接続されている。これは少なくとも、各リージョンのニューロンのうち主要な入力層である第4層のニューロンについて報告されている。

## 末梢樹状突起<sup>65</sup>

細胞体から遠くにある樹状突起の枝は末梢樹状突起と呼ばれる。図では末梢樹状突起のいくつかを青色の線で示した。

---

<sup>61</sup> axon

<sup>62</sup> proximal dendrite

<sup>63</sup> depolarization。神経細胞内の電位は通常は-70~-60mV程度。ニューロンが刺激を受けたためにこの電位が上がることを脱分極という。脱分極により電位があるしきい値を超えるとニューロンが発火する。

<sup>64</sup> action potential。なんらかの刺激に応じて細胞膜に生じる一過性の膜電位の変化。

<sup>65</sup> distal dendrite

末梢樹状突起は主要樹状突起よりも細い。これらは樹状突起の木の中の他の樹状突起の枝に接続されていて、細胞体に直接接続されていない。これらの違いにより末梢樹状突起はユニークな電気・化学特性を持っている。末梢樹状突起で一つのシナプスがアクティブになっても、細胞体に対して最小限の影響しか及ぼさない。シナプスで局所的に発生した脱分極は、それが細胞体に届くときには弱くなっている。このことは長年のなぞであった。ニューロンのシナプスの大多数を占める末梢のシナプスはあまり多くのことをしていないように思えた。

今では末梢樹状突起の各断片が半独立の処理領域として働くことが分かっている。もしその樹状突起の短い区間内で十分な数のシナプスが同時にアクティブになると、樹状突起のパルスを生成することができ、それは細胞体にまで届いて大きな影響を及ぼすことができる。例えば  $40\mu\text{m}$  間隔の 20 個のアクティブなシナプスは樹状突起のパルスを生成するだろう。

従って、末梢樹状突起は域内同時発生事象の検出器のように働くと言える。

末梢樹状突起上のシナプスは、圧倒的にそのリージョン内の付近の他のセルから形成されている。

画像では上方向に伸びる大きな樹状突起の枝が見られる。これは先端樹状突起<sup>66</sup>と呼ばれる。ある理論によると、この構造によりニューロンは付近にいくつかの末梢樹状突起を形成し、この付近を通る軸索により容易に接続することができるようにしているという。この解釈によれば、先端樹状突起はセルの延長として働くと考えられる。

## シナプス

標準的なニューロンには数千個のシナプスがある。これらの大多数(多分 90%)は末梢樹状突起にあり、残りは主要樹状突起にあると思われる。

長年に渡って、学習はシナプスの影響度ないし「重み」を強くしたり弱くしたりすることを含むものと仮定されてきた。このような影響は観測されるものの、各シナプスはどこか確率的な様子である。アクティブになったとき、それが神経伝達物質<sup>67</sup>を放出することに信頼性がない。よって脳が用いているアルゴリズムが各シナプスの重みの精度や忠実度に依存しているはずがない。

さらに今では、シナプス全体が素早く形成されたり切断したりすることが分かっている。この柔軟性は学習の強力な表現形式であり、素早く知識を獲得できることをよりよく説明している。シナプスは軸索と樹状突起がある距離の範囲内にあるときにだけ形成されうることから、シナプス「候補」の概念が得られた。これらの仮定から、学習は主にシナプス候補から有効なシナプスが形成されることで行われると言える。

<sup>66</sup> apical dendrite. apical は「頂点の、頂上の」という意味。

<sup>67</sup> neurotransmitter

## ニューロンの出力

ニューロンの出力は軸索に沿って伝搬するパルス<sup>68</sup>あるいは「活動電位」である。軸索はセルから伸びて、ほとんど常に二つに分かれる。枝の一つは水平に伸びて近くの他のセルと数多く接続する。他の枝は他の層のセルや脳内の他の場所へと遠く伸びる。上記のニューロンの画像では軸索は見えない。2本の矢印を追加して軸索を示した。

ニューロンの実際の出力は常にパルスであるが、この解釈には異なる見方ができる。有力な見方としては（特に新皮質に関しては）、パルスの発生頻度が重要というものである。よってセルの出力はスカラー値と見なすことができる。

いくつかのニューロンは数個の連続したパルスを短時間に素早く出力する「バースト」反応を示すこともあり、これは通常のパルスのパターンとは異なる。

ニューロンに関する上記の記述は、ニューロンを手短に説明したつもりである。これは HTM セルの特徴に関連する属性に注目して述べたもので、多くの詳細は無視している。ここで述べたすべての特徴が幅広く受け入れられているとは必ずしも言えない。これらをここで述べたのは、我々のモデルにとって必要だからである。ニューロンについて知られていることを述べるなら有に数冊の本にもなり、ニューロンの活発な研究は現在も続けられている。

## 単純な人工ニューロン

この付録の最初に示した中央の図は、多くの古典的な人工ニューラルネットワークモデルで使われているニューロンに模した構成要素を示している。これらの人工ニューロンはシナプスの集合を持ち、各シナプスはウェイトを持っている。各シナプスはスカラー値のアクティブ化を受け取り、それにシナプスのウェイトが掛け合わされる。すべてのシナプスの出力は非線形の方法で足し合わされ、人工ニューロンの出力となる。学習はシナプスのウェイトを調整することで行われ、その調整は恐らく非線形の関数となる。このタイプの人工ニューロン、そしてそのバリエーションは、コンピュータを利用した価値ある道具として多くのアプリケーションにおいて有益であることが実証されてきた。しかしながら、それは生体ニューロンの多くの複雑さを捉えておらず、その多くの処理能力を活用していない。実物のニューロンが脳内でどのような全体効果を生み出すかを理解しモデル化したいのなら、我々はもっと精巧なニューロンモデルが必要である。

---

<sup>68</sup> spike。短時間の尖った波形のこと。pulse とは少し違うが、パルスと訳した。

## HTM セル

図の右側は、HTM 大脳皮質性学習アルゴリズムで用いられているセルを図示している。HTM セルは実物のニューロンの多くの重要な能力を捉えているが、いくつかの単純化も行っている。

### 主要樹状突起

各 HTM セルは単一の主要樹状突起を持つ。セルへのすべてのフィード・フォワード入力はシナプス（緑色の点で示した）経由で入力される。アクティブなシナプスの効果は線形に加算され、フィード・フォワード入力によるセルのアクティブ化を生み出す。

我々は、カラム内のすべてのセルが同じフィード・フォワード応答を持つよう求めている。実物のニューロンではこれは恐らく抑制タイプのセルによって行われる。HTM では我々は単にカラム中のすべてのセルが単一の主要樹状突起を共有するように強制することにした。

隣のセルとの競合に決して勝つことができないようなセルができることを避けるため、HTM セルはその隣と比較して十分に勝利していないときには、そのフィード・フォワード入力によるアクティブ化をブーストする。よってセル間には常に競合が存在する。ここでも、我々はこれをセル間ではなくカラム間の競合として HTM でモデル化した。この競合は図では示されていない。

最後に、主要樹状突起はリージョンへのすべての入力の部分集合となるシナプス候補の集合を一つ持っている。セルが学習すると、その主要樹状突起上のすべてのシナプス候補の「永続」値を増加ないし減少させる。しきい値を超えたシナプス候補だけが有効となる。

既に述べたように、シナプス候補の概念は生物学からもたらされた。生物学では、シナプスを形成するのに十分に近くにある軸索と樹状突起とを意味する。我々はこの概念を拡張して、HTM セルで接続する可能性のある、より大きな集合を意味することとした。生体ニューロンの樹状突起と軸索は学習によって成長したり縮退したりすることができ、よってシナプス候補の集合は成長に伴って変化する。HTM セルのシナプス候補の集合を大きめにすることで、我々は概ね、軸索や樹状突起の成長と同じ結果を得た。シナプス候補の集合は図示されていない。

カラム間の競合、シナプス候補の集合による学習、あまり活用されていないカラムのブーストの組み合わせにより、HTM ニューロンのリージョンは脳に見られるのと同様の強力な可塑性を得た。HTM リージョンは入力の変化に応じて各カラムが何を表す

のかを（主要樹状突起上のシナプスの変更によって）自動的に調整し、カラム数の増加ないし減少を自動的に調整するだろう。

## 末梢樹状突起

各 HTM セルは末梢樹状突起セグメントのリストを管理している。各セグメントはしきい値検出器として働く。任意のセグメント上のアクティブなシナプス（冒頭の図で青い点で示した）の数がしきい値を超えると、そのセグメントはアクティブになり、それに接続されたセルが予測状態になる。セルの予測状態はアクティブなセグメントの OR である。

樹状突起セグメントは、以前のある時点で互いに同時にアクティブになった複数のセルへの接続を形成することでリージョンの状態を記憶する。そのセグメントは、以前にフィード・フォワード入力によってアクティブになったセルの状態を記憶する。よってそのセグメントはセルがアクティブになることを予測する状態を探す。樹状突起セグメントの標準的なしきい値は 15 である。あるセグメント上の有効なシナプスが 15 個同時にアクティブになると、その樹状突起はアクティブになる。数百から数千個の周囲のセルがアクティブになるかも知れないが、15 個の接続だけで十分により大きなパターンを理解することができる。

各末梢樹状突起セグメントはそれに関連付けられたシナプス候補の集合を持つ。そのシナプス候補の集合はリージョン内のすべてのセルの部分集合である。そのセグメントが学習するに従い、それらすべてのシナプス候補の永続値を増加ないし減少させる。しきい値を超えたシナプス候補だけが有効である。

ある実装では、我々はセルあたり固定の数の樹状突起セグメントを用いた。他のある実装では、訓練を通じてセグメントを追加ないし削除するようにした。両方の方法が機能する。セルあたりの樹状突起セグメントの数を固定にすると、同じセグメントに対していくつかの異なるシナプスの集合を保存することができる。例えば、セグメント上に 20 個の有効なシナプスがあり、しきい値が 15 とする。（一般に我々はノイズへの耐性を改善するためしきい値をシナプスの数よりも少なくしたい。）これでそのセグメントは周囲のセルの特定の一つの状態を理解できるようになる。もし周囲のセルのまったく異なる状態を表現する、20 個の他のシナプスをその同じセグメントに追加したら何が起こるだろうか。するとそのセグメントは、あるパターンの 8 個のアクティブなシナプスと他のパターンの 7 個のアクティブなシナプスを持つことで間違っただけでアクティブになるかも知れないので、エラーが起こる可能性がもたらされる。我々は実験的に、20 個の異なるパターンまでならエラーなしで一つのセグメントに保存できると分かった。従って十数個の樹状突起セグメントを持つ HTM セルは多くの異なる予測に関与することができる。

## シナプス

HTM セルのシナプスは二値の重みを持つ。HTM モデルのシナプスの重みをスカラー値にすることを妨げるものは何もないが、疎分散パターンを用いることにより、我々は今のところスカラー値の重みを使う必要性がない。

しかしながら、HTM セルは「永続値」というスカラー値を持ち、これを学習を通じて調整する。永続値 0.0 は有効でないシナプス候補を表し、それが有効なシナプスに至るまでまったく進展していないことを表す。しきい値（標準的には 0.2）を超える永続値は接続したばかりで容易に切断するシナプスを表す。高い永続値、例えば 0.9 は、接続状態でしかも容易には切断しないシナプスを表す。

HTM セルの主要樹状突起セグメントや末梢樹状突起セグメントにある有効なシナプスの数は固定ではない。それはセルがパターンに触れるに従って変化する。例えば、末梢樹状突起の有効なシナプスの数はデータの時間的構造に依存する。リージョンへの入力に時間的に永続的なパターンが何もないときは、末梢セグメントのすべてのシナプスは低い永続値を持ち、わずかな数のシナプスだけが有効になるだろう。入力列にたくさんの時間的構造があるときは、高い永続値を持つ有効なシナプスが多数見られるだろう。

## セル出力

HTM セルは 2 つの異なる二値出力を持つ： 1) セルがフィード・フォワード入力によって（主要樹状突起経由で）アクティブである、2) セルが横方向の接続により（末梢樹状突起経由で）アクティブである。前者を「アクティブ状態」と呼び、後者を「予測状態」と呼ぶ。

冒頭の図では、この 2 つの出力は正方形の細胞体から出ている 2 つの線で表されている。左側の線はフィード・フォワードによるアクティブ状態、右側の線は予測状態である。

フィード・フォワードによるアクティブ状態だけがリージョン内の他のセルに接続され、これにより予測は常に現在の入力（及び文脈）に基づいて行われる。予測に基づいて予測が行われることは望ましくない。もしそうになると、数回処理を繰り返すだけでリージョン内のほとんどすべてのセルが予測状態になるだろう。

リージョンの出力はすべてのセルの状態を表すベクトルである。もし階層構造の次のリージョンがあるなら、このベクトルがその入力となる。この出力はアクティブ状態と予測状態の OR である。アクティブ状態と予測状態を結合することで、我々のリージョンの出力は入力よりも安定する（ゆっくりと変化する）。このような安定性はリージョンの推論における重要な特性である。

## 参考文献

ニューロサイエンスをより深く学ぶための参考文献について我々はよく尋ねられる。ニューロサイエンスの分野はたいへん膨大で、全般的知識を得るには多くの異なる文献にあたらねばならない。新しい発見は学術ジャーナルとして出版されるが、読み解くのが難しく大学関係者でない限り入手も難しい。

この付録で述べた話題に関してより専門的知識を得たい読者のために、入手可能な本を2冊示す。

Stuart, Greg, Spruston, Nelson, Hausser, Michael, *Dendrites, second edition* (New York: Oxford University Press, 2008)

この本は樹状突起に関するあらゆることの良い情報源である。16章ではHTM 大脳皮質性学習アルゴリズムが用いている樹状突起セグメントの非線形な性質について述べられている。この章は、この分野で数多くの考察をした Bartlett Mel によって書かれている。

Mountcastle, Vernon B. *Perceptual Neuroscience: The Cerebral Cortex* (Cambridge, Mass.: Harvard University Press, 1998)

この本は新皮質に関するあらゆることに関する良い入門書である。いくつかの章では細胞の種類とその接続関係について述べている。樹状突起の性質に関する最新の知識を幅広く得るには古すぎるが、読者は皮質性ニューロンに関するすぐれた見識が得られるだろう。

## 付録 B: 新皮質の層と HTM リージョンの比較

---

ここでは HTM リージョンと生体新皮質のリージョンの間関係について述べる。

特に、HTM 大脳皮質性学習アルゴリズム、及びそのカラムとセルが、新皮質の層やカラム構造とどのような関係にあるかについて述べる。新皮質の「層」の概念やそれが HTM の層とどう関係するのかについて、多くの人が困惑している。本稿がこの混乱を解決し、また HTM 大脳皮質性学習アルゴリズムの生物学的基礎に関して読者がより深い見識を得られれば幸いである。

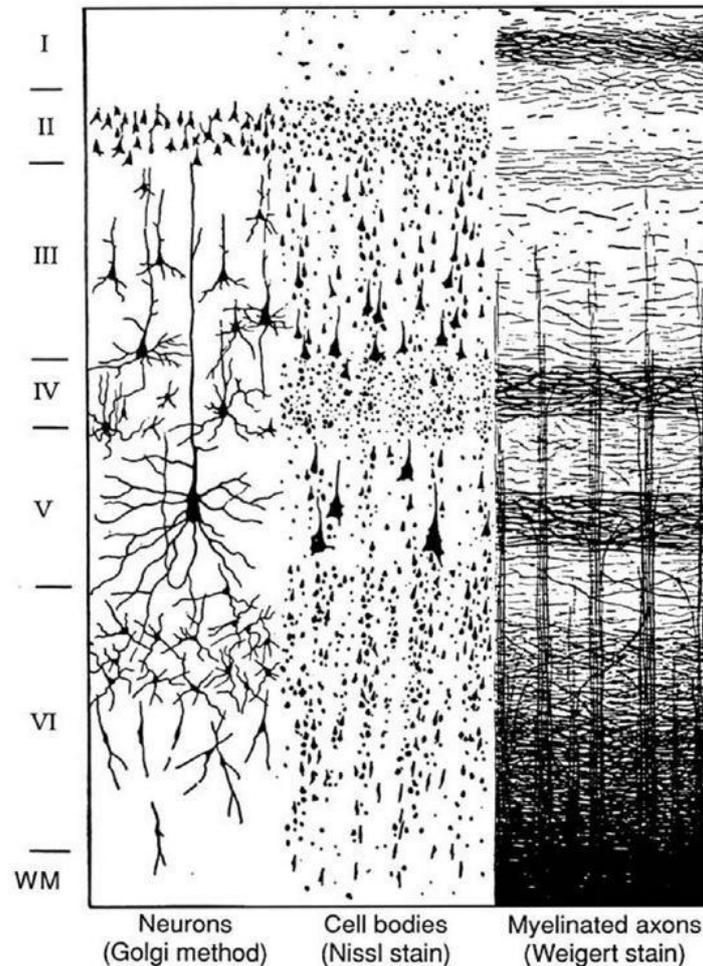
### 新皮質の神経回路網

人の新皮質は面積約 1,000cm<sup>2</sup>、厚さ 2mm のニューロンの皮である。この皮を視覚化するには、食事に使うナプキンの布と考えてみれば、新皮質の面積と厚さのちょうど良い近似になっている。新皮質は十数種類の機能的なリージョンに分かれていて、そのいくつかは視覚に関係し、あるいは聴覚、言語などに関係している。顕微鏡で見ると、異なるリージョンの物理的な特徴は驚くほど良く似ている。

新皮質全体を通じて各リージョンには器官原理<sup>69</sup>がいくつか見られる。

---

<sup>69</sup> organizing principles。生体器官の働きの原理的しくみ。



## 層

新皮質は一般に6つの層を持つと言われている。それらの層のうち5つはセルを持ち、1つの層はほとんどが接続線である。層は染色技術の出現と共に100年以上前に発見された。上記の画像（Cajalによる）は3種類の異なる染色法を用いて新皮質の小さな断片を示したものである。垂直方向の軸索は約2mmの新皮質の厚さ全体に及んでいる。画像の左側は6つの層を示している。最上部の第1層はセルがない層である。最下部の「WM」は白質が始まる場所を示しており、セルからの軸索はそこから新皮質の他の部分や脳他の部分へと伸びている。画像の右側は髄鞘を持つ軸索だけを示す染色法である。（髄鞘形成<sup>70</sup>とは一部の軸索を覆っている脂肪質の鞘<sup>71</sup>である。ただしすべての軸索を覆っているのではない）この部分の画像から新皮質の2つの主要な器官原理である、層とカラムを見ることができる。多くの軸索はニューロンの本体から出た直後

<sup>70</sup> myelination。ずいしょうけいせい。ミエリン化。

<sup>71</sup> sheath。さや。

に2つに枝分かれする。枝の一つは主に水平に伸び、他の枝は主に垂直に伸びる。水平の方の枝は同じ層や近くの層の他のセルと多数の接続を成し、そのために染色でこのように層が見られる。これは新皮質の断片を表していることを心に留めておいて欲しい。多くの軸索がこの画像で示された部分から出たり入ったりしているので、軸索は画像に見られるものよりも長い。新皮質の1mmの立方体に含まれる軸索や樹状突起の総延長は2kmから4kmに及ぶと見積もられている。

画像の中央部はニューロンの本体だけを示す染色法で、樹状突起や軸索は見えない。ニューロンの大きさや密度が層によって変化する様子を見ることができる。この画像ではカラムは少ししか分からない。第1層にいくつかのニューロンがあることに気づかれたかも知れない。でも第1層のニューロンの数はあまりに少ないので、この層はやはりセルのない層と呼ばれている。ニューロ科学者は新皮質の1mmの立方体ごとに約100,000個程度のニューロンがあると見積もっている。

画像の左側はほんのわずかな数のニューロンの本体、軸索、樹状突起だけを示す染色法である。異なる層や異なるセルごとに、樹状突起の「主軸」の大きさは異なっている様子を見ることができる。いくつかの「先端樹状突起」<sup>72</sup>も見られ、それは細胞体からそびえ立ち、他の層と接続している。先端樹状突起が存在するか否か、及びその接続先は各層ごとに特徴がある。

まとめると、新皮質の層とカラム構造<sup>73</sup>はニューロンの皮が染色され顕微鏡で観察されることで明らかになった。

## リージョンの違いによる層のバリエーション

新皮質のリージョンの違いによって層の厚さにバリエーションがあり、層の数についても多少違う。このバリエーションはどの動物を研究するかに依存し、どのリージョンを観察するかにも依存し、また観察した人によっても違う。例えば上記の画像では、第2層と第3層は容易に識別できるが一般的にはそうではない。いくつかの科学者は彼らが研究したリージョンではこの2つの層を識別できないと報告しており、第2層と第3層をまとめて「第2/3層」と呼ぶこともしばしばある。他の科学者は逆の方向に向かい、例えば3Aと3Bのようなサブレイヤを定義している。

第4層は、新皮質のリージョンの中で感覚器官にもっとも近い部分で最もよく定義されている。いくつかの動物（例えばヒトやサル）では、第1視覚野の第4層は明確に細分化されている。他の動物ではそれは細分化されていない。第4層は感覚器官から遠いリージョンでは階層構造から消えて無くなっている。

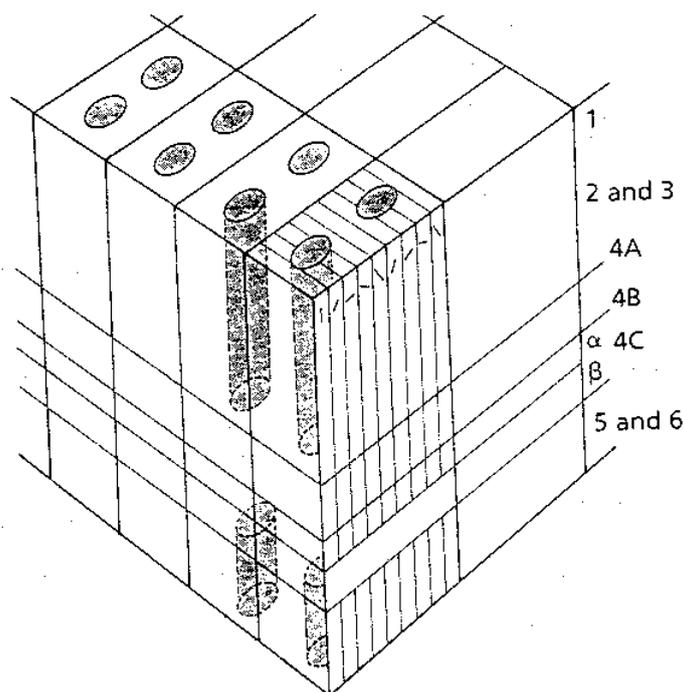
<sup>72</sup> apical dendrite

<sup>73</sup> columnar organization. 一般には柱状構造。他と統一するためカラム構造とした。

## カラム

新皮質の2つ目の主要な器官原理はカラムである。いくつかのカラム構造は染色された画像にも見られるが、カラムに関する多くの証拠は異なる入力に対してセルがどのように反応するかに基づいている。

科学者が針を使って、何がニューロンをアクティブにするのかを見てみると、異なる層を横断する垂直方向に揃った複数のニューロンがだいたい同じ入力に反応することを見つけた。



この図は、網膜からの情報を処理する最初の皮質性リージョンである V1 の、セルのいくつかの応答特性を示している。

最初の発見の一つは、V1 のほとんどのセルは網膜の特定の領域で、異なる角度の線や縁に反応するということであった。カラム状に垂直に配列された複数のセルすべてが、同じ角度の縁に反応する。図を注意深く見れば、各区画の最上部に異なる角度の小さな線が描かれていることが分かるだろう。これらの線はその場所のセルがどの角度の線に反応するかを示している。垂直に配列された複数のセル（うすい垂直の縞模様の一つに含まれる）は同じ角度の線に反応する。

V1 にはいくつかの他のカラム型の特徴があり、そのうちの2つが図示されている。左目と右目の情報の似た組み合わせにセルが反応する「眼球優位性カラム」<sup>74</sup>がある。

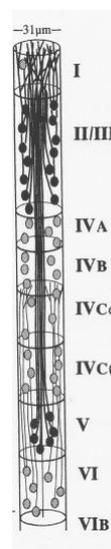
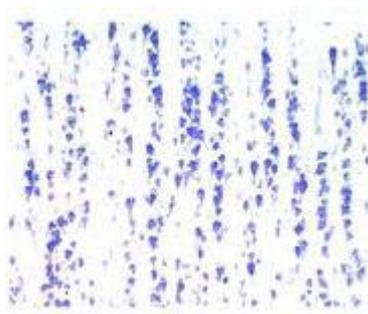
<sup>74</sup> ocular dominance column。片方の眼からの入力に強く反応するセルの集まり。

そしてセルが主に色を感知する「ブロブ」<sup>75</sup>がある。眼球優位性カラムは図の大きなブロックである。各眼球優位性カラムは角度のカラムを含む。「ブロブ」は濃い色の楕円である。

新皮質の一般的な規則は、角度と眼球優位性のよういくつかの異なる応答特性が互いに重ね合わさっているということである。皮質の表面を水平に移動してゆくに従って、セルから出力される応答特性の組み合わせは変化する。しかしながら、垂直に配列されたニューロンは同じ応答特性の組み合わせを共有している。聴覚・視覚・体性感覚野についてはこのような垂直の配列になっている。新皮質のあらゆる場所でそうになっているのかについては科学者達の間でいくつかの議論があるが、全部ではなく多くの部分について言うならそれは正しいようである。

### ミニカラム

新皮質の最小のカラム構造はミニカラムである。ミニカラムは直径約  $30\mu\text{m}$  で、セルを持つ5つの層全体に及ぶ80-100個のニューロンが含まれている。新皮質全体はミニカラムから構成されている。小さなスパゲッティのかけらを端同士を積み重ねたものを思い浮かべるとよい。ミニカラムの間にはセルが少ししかないわずかなすきまがあり、染色された画像でそれを見ることができる。



左側は新皮質の一部の断片に見られるニューロンの細胞体を示す染色画像である。ミニカラムの垂直の構造がこの画像から明白に分かる。右側はミニカラムの概念図である (Peters と Yilmaz による)。実際にはこれよりずっと細い。カラムの中の各層に複

<sup>75</sup> blob。小塊、小球体。

数のニューロンがあることに注意してほしい。ミニカラムのすべてのニューロンが類似の入力に反応する。例えば、先ほど示した V1 の図では、ミニカラムは特定の眼球優位性を伴い、特定の角度の線に反応するセルを含んでいる。隣にあるミニカラムのセルは少し違う角度の線に反応し、違う眼球優位性を示すのかも知れない。

抑制ニューロンがミニカラムを定義する本質的な役割を果たしている。それらは画像や図に示されていないが、抑制ニューロンはミニカラムの間のまっすぐな線に沿って軸索を送っており、ミニカラムの一部を物理的に分離している。抑制ニューロンはまた、ミニカラム中のニューロンが同じ入力に反応するよう強制することに役立っていると信じられている。

ミニカラムは HTM 大脳皮質性学習アルゴリズムで用いられているカラムの原型である。

## カラム反応の例外

カラム反応の例外が一つあって、それは HTM 大脳皮質性学習アルゴリズムにも関係する。科学者は通常、実験動物に単純な刺激を与えることでセルが何に反応するのかを発見する。例えば、動物の視覚空間の小さな部分に 1 つの線を見せて、V1 のセルの応答特性を調べようとするかも知れない。単純な入力を用いると、科学者はセルが常に同じ入力に反応することを発見するかも知れない。しかしながら、もしその単純な入力が自然な場面の動画像に組み込まれたなら、セルはもっと選択的になる。あるセルが高い信頼性で独立した垂線に反応するとしても、その垂線が自然な場面の複雑な動画像に組み込まれた場合は必ずしも反応するとは限らない。

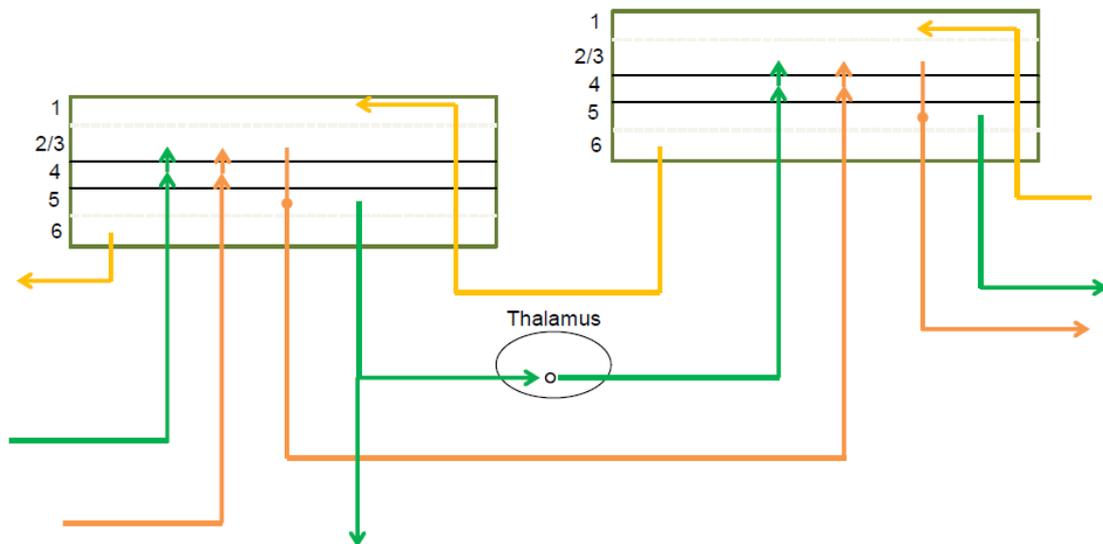
HTM 大脳皮質性学習アルゴリズムではカラム中のすべての HTM セルが同じフィード・フォワード応答特性を共有しているが、時間的なシーケンスを学習すると HTM カラムの 1 つのセルだけがアクティブになる。このメカニズムは可変長シーケンスを表現する手段であり、ニューロンについて先ほど説明した特徴と似ている。文脈を伴わない単純な入力はカラム中のすべてのセルをアクティブにする。同じ入力でも、学習したシーケンスに含まれるときは 1 つのセルだけがアクティブになる。ミニカラム中で一度に 1 つのニューロンだけがアクティブになると提唱しているわけではない。HTM 大脳皮質性学習アルゴリズムが提唱しているのは、予期しない入力に対してはカラム中のある層のすべてのニューロンがアクティブになり、予期した入力に対してはその一部のニューロンがアクティブになるということである。

## なぜ層とカラムがあるのか？

新皮質になぜ層があり、なぜカラムがあるのか、はっきりしたことは誰も知らない。HTM 大脳皮質性学習アルゴリズムは、カラム状に構成したセルの層が可変長の状態遷移を記憶する大容量メモリとなりうることを示した。もっと単純に言えば、セルの層はたくさんのシーケンスを学習できるということである。同じフィード・フォワード反応を共有するセルのカラムは可変長の遷移を学習するための鍵となるメカニズムである。

この仮説はなぜカラムが必要なのかを説明しているが、しかし5つの層についてはどうだろうか？ もし1層の皮質でシーケンスを学習して予測できるのであれば、なぜ新皮質に5つの層が見られるのであろうか？

我々が提唱するのは、新皮質に観察される異なる層はすべて同じ基本メカニズムを用いてシーケンスを学習しているが、各層で学習したシーケンスは異なる方法で使用されているのだらうというものである。これについて我々が理解していないことはたくさんあるが、我々の一般的な考えを述べることはできる。その前に、各層のニューロンが何に接続しているのかを述べると理解の助けとなるだろう。



上の図は2つの新皮質のリージョンとそれらの間の主要な接続関係を示している。このような接続は新皮質の中の互いに関係し合う2つのリージョンで一般によく見受けられる。左の箱は、右の(箱の)リージョンよりも低い階層構造にある皮質性のリージョンを表しているので、フィード・フォワード情報は図の左から右へと流れる。各リージョンは層に分けられている。第2層と第3層は一緒にして第2/3層として表されている。

色のついた線は異なる層からのニューロンの出力を表している。これらはその層のニューロンから出ている軸索の束である。軸索はすぐに2つに分かれることを思い出して欲しい。一つの枝は主にそれと同じ層の中で、リージョン内で水平方向に広がる。よって各層のすべてのセルは相互によく接続し合っている。ニューロンと水平方向の接続は図に示されていない。

2つのフィード・フォワード・パス<sup>76</sup>がある。オレンジ色で示した直接パスと、緑色で示した間接パスである。第4層は主要なフィード・フォワード入力層で、両方のフィード・フォワード・パスから入力を受け取る。第4層は第3層に向かう。

第3層は直接フィード・フォワード・パスの始点でもある。よって、直接フィード・フォワード・パスは第4層と第3層に限定されている。

いくつかのフィード・フォワード接続は第4層を飛ばして直接第3層に至る。そして、上記で述べたように、第4層はセンサ入力から遠くにあるリージョンでは消えて無くなっている。そこでは直接フォワード・パスは単に第3層から次のリージョンの第3層に繋がっている。

2つめのフィード・フォワード・パス（緑色で示した）は第5層から始まる。第3層のセルは次のリージョンに至る道筋の中で第5層へと接続している。皮質性の皮から出発した後、第5層のセルからの軸索は再び枝分かれする。1つの枝は運動の生成に関わる脳内の皮質下部<sup>77</sup>へと向かう。これらの軸索は運動指令（下方向の矢印で示した）であると信じられている。他の枝は脳内の視床<sup>78</sup>と呼ばれる門として働く部分へと向かう。視床は次のリージョンに情報を通したり止めたりする。

最後に、黄色で示した主要フィードバック・パスが第6層から第1層に向かっている。第2, 3, 5層のセルは先端樹状突起（図に示されていない）を経由して第1層に向かっている。第6層は第5層から入力を受け取る。

この説明は層から層への接続に関して知られていることを限定的に概説したものである。しかし、すべての層がシーケンスを学習するのになぜ複数の層が存在するのかについての我々の仮説を理解するには十分であろう。

## 異なる層が何をするのかに関する仮説

我々は第3, 4, 5層がすべてフィード・フォワード層でありシーケンスを学習していると提唱した。第4層は一次シーケンスを学習する。第3層は可変長シーケンスを学習

---

<sup>76</sup> pathway。通り道。

<sup>77</sup> sub-cortical area。大脳皮質の下の神経中枢。

<sup>78</sup> thalamus。ししよ。

する。第5層はタイミングを含む可変長シーケンスを学習する。これらのそれぞれについて、より詳しく見ていこう。

#### 第4層

HTM 大脳皮質性学習アルゴリズムを用いて一次シーケンスを学習するのは容易である。もしカラム中のセルが互いに抑制するように強制しなかったとしたら、つまりカラム中のセルが以前の入力の文脈を区別しなかったとしたら、一次学習が起こる。新皮質では、同じカラム内のセルを抑制する効果を取り除くことで成されるだろう。我々のコンピュータモデルである HTM 大脳皮質性学習アルゴリズムでは、単にカラムごとに1つのセルを割り当てることで同様の結果を生む。

一次シーケンスは入力空間的変形<sup>79</sup>を表す不変表現<sup>80</sup>を作る上で必要とされるものである。例えば視覚では、 $x$ - $y$  変換、縮尺、回転はすべて空間的変形である。移動する物体について、一次の記憶を持つ HTM リージョンを訓練すると、異なる空間的パターンが同等であることを学習する。結果の HTM セルは「複雑型細胞」と呼ばれるもののように振舞う。その HTM セルはある範囲の空間的変形に対してアクティブな状態（予測状態）を保つ。

Numenta では、視覚についてこのメカニズムが期待通りに機能することを検証する実験を行い、いくつかの空間的不変性が各レベルで達成された。これらの実験の詳細はこの付録の範囲を超える。

第4層で一次シーケンスを学習していることは、第4層で複雑型細胞が見られることや、なぜ新皮質の高階層のリージョンで第4層が消えて無くなるのかということと符合している。階層構造を上がるにつれて、その時点での表現はすでに不変のものになっているためそれ以上空間的不変性を学習することはできなくなる。

#### 第3層

第3層は第2章で述べた HTM 大脳皮質性学習アルゴリズムに最も近い。それは可変シーケンスを学習し、予測を行い、その予測は入力よりも安定している。第3層は常に階層構造の次のリージョンに向かい、そのため階層構造の中で時間的安定性がより増加するようになる。可変シーケンス記憶は「方位選択性複雑型細胞」<sup>81</sup>と呼ばれるニューロンに形成され、それは第3層で最初に観察された。方位選択性複雑型細胞は例えば左に動いている線と右に動いている線など、時間的文脈による識別をする。

---

<sup>79</sup> spatial transformation

<sup>80</sup> invariant representation

<sup>81</sup> directionally-tuned complex cell

## 第5層

最後のフィード・フォワード層は第5層である。我々は第5層が第3層と似ているが3つの違いがあると提唱する。第一の違いは第5層が時間の概念を付加するということである。第3層は次に「何」が起こるかを予測するが、それが「いつ」起こるかを教えてくれない。しかしながら、話し言葉を理解するときには音の間の相対的なタイミングが重要であるように、多くのことでタイミングが必要となる。別の例として運動動作がある。筋肉の活性化のタイミングを揃えることは本質的である。我々は、第5層のニューロンが期待した時刻の後にだけ次の状態を予測すると提唱する。この仮説を裏付ける生物学上の詳細がいくつかある。一つは第5層が新皮質の運動出力層であるということである。いま一つは第5層が視床の一部から発して第1層から来る入力(図に示されていない)を受け取るということである。我々はまた、この情報こそが時間をコード化したものであり、視床を経由して第1層に入力される多くのセル(図に示されていない)にこの情報が分散されると提唱する。

第3層と第5層の間の第二の違いは第3層が可能な限り未来を予測して時間的安全性をもたらすことが望ましいということである。第2章で述べたHTM 大脳皮質性学習アルゴリズムはこれを行う。対照的に、第5層については次の項目(ある特定の時点の)を予測することしか求めていない。我々はこの違いをモデル化していないが、遷移が常に時間を伴って保存されるならそれは自然に起こる。

第3層と第5層の間の第三の違いは図から見て取れる。第5層の出力は常に皮質下部の運動中枢に向かい、そのフィード・フォワード・パスは視床の門を通る。第5層の出力は、あるときは次のリージョンへと通過し、またあるときは止められる。我々(及び他の人)はこの門の働きが潜在的注意<sup>82</sup>に関係すると提唱する(潜在的注意は運動行動を伴わずに貴方が入力に注目することである)。

まとめると、第5層は特定のタイミング、注意、運動行動を結びつける。これらが互いにどのように関わりあうかについては多くの謎が残されている。我々が言いたいポイントは、HTM 大脳皮質性学習アルゴリズムのバリエーションが特定のタイミングを容易に組み入れることができ、別々の皮質の層を結合することができるということである。

## 第2層と第6層

第6層は下位のリージョンへフィードバックする軸索の起点である。第2層についてはほとんど知られていない。上で述べたように、第2層が第3層と比べてユニークな点があるかどうかですら、しばしば議論となっている。我々はいまのところこの質問に関してこれ以上言えることはほとんどないが、他のすべての層と同様に第2層と第6層はたくさんの水平方向の接続パターンを持ち、カラム単位で反応する特徴があること

---

<sup>82</sup> covert attention

だけは指摘することができる。よって我々はこれらもまた、HTM 大脳皮質性学習アルゴリズムの一形態を実行していると提唱する。

## HTM リージョンは新皮質の何に相当するか？

我々は 2 種類の HTM 大脳皮質性学習アルゴリズムを実装した。一方は可変長記憶のためにカラムごとに複数のセルを持たせるもので、他方は一次記憶のためにカラムごとに単一のセルを持たせるものである。我々はこの 2 種類が新皮質の第 3 層と第 4 層に相当すると信じる。これら 2 種類を単一の HTM リージョンに結合することを我々はまだ試みていない。

HTM 大脳皮質性学習アルゴリズム（カラムごとに複数のセルを持つ）が新皮質の第 3 層に最も近いものの、我々のモデルは脳にもない柔軟性を持っている。よって我々は新皮質のどの層にも相当しない複合型のセルを持つ層を創ることができる。例えば、我々のモデルでは樹状突起セグメント上でシナプスが形成される順序が分かる。我々はこの情報を使って、将来起こることのすべてをより一般的に予測した上で次に何が起こるかを予測することができる。我々は多分、同様にしてタイミング特有のことを追加できるだろう。従って単一の層の HTM リージョンに第 3 層と第 5 層の機能を結合したものを作ることが可能だろう。

## まとめ

HTM 大脳皮質性学習アルゴリズムは我々が新皮質の神経器官の基本構成要素と信じているものを具現化するものである。それは水平接続されたニューロンの層がどのようにして疎分散表現のシーケンスを学習するのかを示している。HTM 大脳皮質性学習アルゴリズムの各バリエーションが、互いに関連するが異なる目的を持つ、新皮質の異なる層で使われる。

我々は新皮質リージョンへのフィード・フォワード入力は、第 4 層であれ第 3 層であれ、主要樹状突起に主に入力されると提唱する。それは抑制セルの働きにより、入力の疎分散表現を作成する。我々はまた、第 2, 3, 4, 5, 6 層のセルがこの疎分散表現を共有していると提唱する。このことは、それらの層をまたがるカラム中のすべてのセルが同じフィード・フォワード入力に反応するように強制することによって達成される。

我々は第 4 層のセルが、もしそれが存在するならば、HTM 大脳皮質性学習アルゴリズムを用いて一次の時間的遷移を学習すると提唱する。これは空間的遷移に対して不変の表現を構成する。第 3 層のセルは HTM 大脳皮質性学習アルゴリズムを用いて可変長の時間的遷移を学習し、皮質の階層を上って行って安定した表現を構成する。第 5 層のセルはタイミングを伴う可変長の遷移を学習する。第 2 層と第 6 層については特に提唱

するものはない。しかしながら、これらの層でよく見られる水平接続を考えると、何らかの形でシーケンス記憶を学習していると考えられる。

## 用語の説明

ノート：ここでの定義はこの文書で使われている用語の意味であり、一般的な意味とは異なるものもある。説明文中で大文字で示されたもの<sup>83</sup>は、この用語説明で説明されていることを示す。

アクティブ状態 (Active State)	フィード・フォワード (Feed-Forward) 入力によってセル (Cells) がアクティブになった状態
ボトムアップ (Bottom-Up) <sup>84</sup>	フィード・フォワード (Feed-Forward) と同義語
セル (Cells)	HTMにおいて、ニューロン (Neuron) に対応するもの。セル (Cells) はHTMのリージョンにおいてカラムを構成する。
同時発生アクティビティ (Coincident Activity)	同時に2個又はそれ以上のセルがアクティブになること
カラム (Column)	1個又はそれ以上のセルのグループで、HTMリージョンの中で1単位として機能するもの。カラム中のセルは、同じフィード・フォワード入力の異なる文脈を表現する。
樹状突起セグメント (Dendrite Segment)	シナプスが集約した単位で、セルやカラムに結び付けられる。HTMには二つの異なるタイプの樹状突起セグメントがある。一つは、あるセルの横方向の接続に結び付けられる。樹状突起セグメントのアクティブなシナプスの数がしきい値を超えると、結び付けられたセルが予測状態になる。もう一方は、あるカラムのフィード・フォワード接続に結び付けられる。あるカラムのアクティブなシナプスの数がしきい値を超えると、フィード・フォワードによるアクティブ状態になる。

<sup>83</sup> 用語説明では、例えば Cell の場合「セル(Cell)」のように示した。

<sup>84</sup> 原文では Bottom-Up と Feed-Forward の 2 種類の用語が現れるが、同様の意味であるとされているため、常に「フィード・フォワード」と訳した。

望ましい密度 (Desired Density)	リージョン (Region) へのフィード・フォワード (Feed-Forward) 入力によってアクティブになるカラム (Column) の望ましいパーセンテージ。このパーセンテージは、フィード・フォワード入力のファンアウト <sup>85</sup> に依存して変化する半径 <sup>86</sup> 内にも適用される。パーセンテージは個別の入力に応じて変化するものなのでここでは「望ましい」と呼んでいる。
フィード・フォワード (Feed-Forward)	階層構造 (Hierarchy) の中で、入力が低いレベル (Level) から高いレベル (Level) に向かって移動すること (しばしば、Bottom-Upと呼ぶ)
フィードバック (Feedback)	階層構造 (Hierarchy) の中で、高いレベル (Level) から低いレベル (Level) に向かって移動すること (しばしば、Top-Downと呼ぶ)
一次予測 (First Order Prediction)	過去の入力には無関係に、現在の入力だけに依存して予測すること。可変長予測 (Variable Order Prediction) 参照。
HTM (Hierarchical Temporal Memory)	新皮質の構造的・アルゴリズム的機能のいくつかを模写する技術
階層構造 (Hierarchy)	要素間の接続がフィード・フォワード (Feed-Forward) ないしフィードバック (Feedback) によってユニークに識別されるネットワーク
HTM大脳皮質性学習アルゴリズム (HTM Cortical Learning Algorithms)	空間プーリング (Spatial Pooling)、時間プーリング (Temporal Pooling)、学習と忘却を行う関数一式。HTMリージョン (HTM Region) を構成する。またの名をHTM学習アルゴリズム (HTM Learning Algorithms) と言う。
HTMネットワーク (HTM Network)	HTMリージョン (HTM Region) の階層構造 (Hierarchy)
HTMリージョン (HTM Region)	HTMにおいて、記憶と予測 (Prediction) を行う主要構成要素。HTMリージョン (HTM Region) は、カラムの中に配置された高度に相互接続された層からなる。現状のHTMリージョン (HTM Region) は一層のセルからなるが、新皮質では (そして完璧なHTMでは) リージョンは複数のセルの層からなる。階層構造の中の位置という文脈で呼ばれるとき、リージョンはレベルと呼ばれることがある。

<sup>85</sup> fan-out。広がり具合

<sup>86</sup> radius

推論 (Inference)	空間的ないし時間的入力パターンが、以前に学習したパターンと似ていると認識すること
抑制半径 (Inhibition Radius)	カラム (Column) の周囲の領域で、その範囲でアクティブなカラムが抑制をする範囲を定義する
横方向の接続 (Lateral Connections)	同じリージョン内でのセル (Cells) 間の接続関係
レベル (Level)	階層構造 (Hierarchy) の中の HTMリージョン (HTM Region)
ニューロン (Neuron)	脳内で情報処理を行うセル (Cells)。本書では、特に生物学的な意味でセルを示すときにニューロンという用語を用い、単にセルと表記したときはHTMの計算単位を意味する。
永続値 (Permanence)	シナプス候補 (Potential Synapse) の接続状態を表すスカラー値。永続値がしきい値を下回るときシナプスは形成されていないことを表す。永続値がしきい値を超えていたら、そのシナプスは有効である。HTMリージョン (HTM Region) の学習はシナプス候補 (Potential Synapse) の永続値を変更することで達成される。
シナプス候補 (Potential Synapse)	ある樹状突起セグメント (Dendrite Segment) でシナプスを形成する可能性があるセル (Cells) の部分集合。ある時刻においては、シナプス候補の一部分だけが、有効なシナプスとなる。有効なシナプスは永続値に基づいて決まる。
予測 (Prediction)	フィード・フォワード (Feed-Forward) 入力によって、セル (Cells) が近い将来アクティブになるであろうということを、(予測状態の) アクティブ化によって示すこと。HTMリージョン (HTM Region) はしばしば、将来起こりうる入力を同時に多数予測する。
受容野 (Receptive Field)	カラム (Column) ないしセル (Cells) が接続されている入力の集合。HTMリージョン (HTM Region) への入力がビットの2D配列で構成されていとき、受容野は入力空間内のある半径範囲で表現することができる。
センサー (Sensor)	HTMネットワーク (HTM Network) への入力源
疎分散表現 (Sparse Distributed Representation)	多くのビットで構成され、そのうちのわずかなパーセンテージだけがアクティブであり、単一のビットだけでは意味を表現するには不十分であるような表現。

空間プーリング (Spatial Pooling)	入力に対して疎分散表現を計算する処理。空間プーリングの一つの特徴は、オーバーラップする入力パターンを同じ疎分散表現に対応付けられることである。
サブサンプリング (Sub-Sampling)	大きなパターンのうちのほんのわずかなアクティブビットをマッチングするだけで、大きな分散パターンを認識すること
シナプス (Synapse)	学習によって形成されるセル (Cells) 間の接続
時間プーリング (Temporal Pooling)	入力パターンのシーケンスの表現を計算する処理。結果の表現は入力よりも安定したものになる。
トップダウン (Top-Down)	フィードバック (Feedback) の同義語
可変長予測 (Variable Order Prediction)	それが依存する直前の文脈の量が変化するような予測。一次予測 (First Order Prediction) 参照。 直前の文脈を維持管理するためのメモリを必要に応じて割り当てるため、「可変長」と呼ばれる。そのため可変長予測を実装するメモリシステムは、指数関数的に増大するメモリを必要とすることなく、文脈を時間的に後戻りすることができる。