



HIERARCHICAL TEMPORAL MEMORY

including

HTM Cortical Learning Algorithms

层级实时记忆 (HTM) 脑皮质学习算法

俞天翔 译

版本 **0.2, 12, 10, 2010**

©Numenta, Inc. 2010

对于非商业目的的研究，使用 Numenta 的软件及知识产权，包括本文档中的观点是免费的。
详情参见：<http://www.numenta.com/about-numenta/licensing.php>.

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.

Numenta 翻译许可

版权所有 2010,2011 Numenta 有限公司

本公司保留所有权利

此处的文字、算法式、样例代码、伪代码、以及其他形式的工作成果均基于 Numenta 有限公司发布的层级实时记忆 (HTM) 技术。Numenta 享有原始成果的版权和涉及到 HTM 及此处算法的专利权。只要是用于研究的非商业的开发和使用, Numenta 同意不追究关于独立的 HTM 系统的使用及开发专利权。任何侵犯 Numenta 专利权的有关 HTM 技术的商业生产行为都需要授权。

基于上述, Numenta 授予你非商业生产仅用于研究目的的算法及相关成果的使用许可。本许可中的“商业生产用途”包括以发布经过训练的商业生产用途的网络和软件为目的而进行的网络训练, 使用或允许他人使用由 HTM 技术得到的结果进行商业生产。任何发布、出版、复制本成果的文档都需要包含完整的英语及目标语言的“翻译许可”。

本许可没有授予关于任何专利权的明确或暗指的许可。Numenta 对于以下得到许可的译文的质量及准确性概不负责。

写在前头

这是本文档的一个草稿版本，你应该知道有一些东西是缺失的。

本文档包括什么

本文档详细描述了由 Numenta 于 2010 开发的新型学习预测算法。看过本算法的介绍的程序员 如果需要能够完全理解和实现本算法。本文档将由一个介绍章节开始，如果你以前读过 Numenta 的其它技术文档，你会发现介绍章节中的内容十分熟悉，但是其他的内容是新的。

本文档不包括什么

有一些涉及到实现这些新算法的话题并没有被写入到这个早期的草稿中。

-虽然此算法的诸多方面已经实现成软件并经过测试，但详细的测试结果并不包括。

-本文档并不包含如何将此算法应用到实际问题中的说明。缺失的说明是如何将数据从传感器或数据库转化成适用于本算法的离散化数据结构。

-你能够在线学习此算法。一些特殊案例中没有涉及的细节你需要通过在线学习来获得。

-其他预计附加的内容包括有关稀疏离散表征的性能的讨论，关于应用程序和样例的介绍，以及附加内容的相关引文。

我们之所以用这样的形式发布是因为我们认为这将使得大多数人对此算法充满兴趣。对于 积极的研究者，缺少的内容并不会对你理解和实验本算法造成阻碍。我们将定期修订本文档 以反映我们的研究进度。

目录

前言	:		4
第一章	:	HTM 概述	7
第二章	:	HTM 脑皮质学习算法	19
第三章	:	空间沉积的实现和伪代码	34
第四章	:	时间沉积的实现和伪代码	39
附录 A	:	生物神经元细胞与 HTM 细胞的对比	47
附录 B	:	新大脑皮层的层与 HTM 中的层的对比	54
术语表			65

前言

有许多事情对于人类而言简单，但是计算机目前是无法做到的。诸如视觉模式识别，理解自然语言，认知以及通过触觉操控物体和在复杂环境中行进等任务对于人类而言是简单的。尽管进行了数十年的研究，我们任然未能在计算机上实现和人类一样的行为能力。

对人类而言，这些能力大多数都是由新大脑皮层控制执行。层级实时记忆（HTM）就是一种模仿新大脑皮层运作的技术。HTM 是构建具有接近或超越人类行为水平的智能机器的保证。

本文档详细描述了 HTM 技术。第一章是关于 HTM 的一个总述，概括了层级结构、稀疏离散表征和基于时间的渐进式学习的相关重点。第二章详细描述了 HTM 脑皮质学习算法。第三、四章介绍了 HTM 学习算法的两部分，空间数据池和时间数据池的实现伪代码。在学习了二至四章的内容后，有经验的软件工程师可以自己编写本算法的程序。我们希望读者们能走的更远并拓展我们的成果。

目标读者

本文档针对的是具有一定技术和教育基础的读者。我们不假定你具有神经科学的知识，但我们认为你能理解一些数学和计算机科学的概念。我们将本文档撰写成可以用作教学的课堂读物。我们第一理想的读者是计算机科学或认知科学的学生，或者是对构建具有和人脑相同原理的智能认知系统感兴趣的软件开发者。

没有基础的读者任然可以从本文档的某些部分受益，尤其是第一章。

软件发布

我们计划在 2011 年年中发布基于本文中算法的软件。

涉及的先前文档

部分 HTM 理论在 2004 出版的《人工智能的未来》一书，Numenta 发布的白皮书以及 Numenta 员工撰写的业内文档中均有所介绍。我们不假设你读过先前的文档，其中的大部分已更新并包括在本文档之中。需要注意的是第二章至第四章的 HTM 学习算法并没有在先前的文档中发表过。这个新的算法替代了我们原先称为 Zeta 1 的第一代算法。我们曾临时称这个算法“固定密度的离散表征” (Fixed-density Distributed Representations)或“FDR”，但我们已经不再使用这个术语。我们称这一新的算法为 HTM 脑皮质学习算法，或者有时简称 HTM 学习算法。

我们鼓励你阅读由 Numenta 公司的创始人 Jeff Hawkins 以及 Sandra Blakeslee 撰写的《人工智能的未来》一书。虽然此书中并未提到 HTM 这一名词，但是它用易读及非技术性的语言描述了 HTM 理论及其相关的神经科学知识。在撰写《人工智能的未来》的时候我们明白了支撑 HTM 的基本原则，但是我们并不知道如何将其算法化。你可以把本文档当做《人工智能的未来》的后续。

关于 Numenta

Numenta 有限公司(www.numenta.com) 始建于 2005 年，致力于 HTM 技术的商业与科学用途的研究工作。为了实现这一目标，我们将探索的成果写成了丰富的文档。我们还将我们的软件以可供他人进行科学与商业开发的形式发布。我们之所以构建这个软件是为了鼓励独立的应用开发团体的出现。以研究为目的使用 Numenta 的软件及相关知识产权是免费的。我们靠技术支持，授予软件许可及知识产权的商业开发许可来获利。我们将一直致力于让我们及我们的开发伙伴获得双赢。

Numenta 位于美国加利福尼亚州的红杉城，它是由私人资助的。

关于作者们

此文档是 Numenta 员工通力合作的结果。每一部分的主要作者将在修订历史记录中给出。

修订历史记录

我们只将有重大改变的版本列举出来，细微的改变将不做记录

版本	日期	改变	主要作者
0.1	Nov 9, 2010	1. Preface, Chapters 1,2,3,4, and Glossary: first release	Jeff Hawkins, Subutai Ahmad, Donna Dubinsky
0.1.1	Nov 23, 2010	1. Chapter 1: the Regions section was edited to clarify terminology, such as levels, columns and layers 2. Appendix A: first release	Hawkins & Dubinsky Hawkins
0.2	Dec 10, 2010	1. Chapter 2: various clarifications 2. Chapter 4: updated line references; code changes in lines 37 and 39 3. Appendix B: first release	Hawkins Ahmad Hawkins

第一章：HTM 概述

层级实时记忆是一种以捕捉新大脑皮层的结构与算法特性为目标的机器学习技术。

新大脑皮层在哺乳动物的大脑中对于智能思维有着重要的位置。高级视觉、听觉、触觉、运动、语言以及计划都是新大脑皮层所控制的。看到如此多的认知功能你或许认为新大脑皮层对于每一种功能都有一种对应的神经算发。但是情况正好相反，新大脑皮层的神经回路具有高度的统一性。生物学研究指出，新大脑皮层用一套公用算发去实现许多不同的智能功能。

HTM 为理解新大脑皮层及其机能提供了一个理论框架。目前为止我们已经实现了这个理论框架的一小部分。随着时间的迁移，关于本理论更多的部分将被实现。我们相信我们至今已实现了足够的 新大脑皮层在商业及研究方面价值的部分内容。

使用 HTM 编程不同于传统的电脑编程。现今的程序员针对特定的问题编写特定的程序。相对的，HTM 通过感知数据流来训练自身。HTM 的机能很大程度上是由训练它的数据所决定的。

HTM 可以被视为一个神经网络。根据定义，任何试图对新大脑皮层架构细节建模的系统都是一个神经网络。然而，就它本身而言，“神经网络”这一术语并不十分适用，因为它被用于许多系统。个神经网络。然而，就它本身而言，“神经网络”这一术语并不十分适用因为它被用于许多系统。HTM 神经元模型（在 HTM 中称为细胞"cell"）是按列,按层,按区域排列的且有等级划分的。这也是为什么把 HTM 看作一种新型神经网络。

顾名思义，HTM 从根本上讲是一个基于记忆的系统。HTM 网络被大量具有时间性的数据训练而成，并依赖于储存的大量模式序列。从逻辑上讲 HTM 的数据存贮访问方式不同于当今程序员使用的标准模型。传统的计算机记忆模型是一种平面的组织结构并且没有考虑时间性。一个程序员可以实现任意的数据组织和结构在计算机内存上。他们能够完全掌控数据怎样储存以及储存到哪。相对而言 HTM 的记忆系统就有很多的限制了。HTM 的记忆是有等级性的并且是基于时间性的。信息总是被分散式的储存。用户可以指定 HTM 的等级划分以及用什么样数据去训练，但是 HTM 控制数据怎样储存以及 储存在哪。

尽管 HTM 网络从实质上讲与传统计算不同，我们仍可以用普通电脑对其进行建模只要我们包含了层级、时间性、稀疏离散表征（稍后介绍）的关键特性。我们相信假以时日，专门用于生成 HTM 网络的硬件将会诞生。

在本文中我们会经常使用一些人类视觉、触觉、听觉、语言及行为的例子来阐明 HTM 的特性和规则。这些例子十分有用因为它们直观并且容易理解。但是我们需谨记 HTM 能做到的远不止这些。它可以很容易的接受来自非人类感官的输入流，例如雷达和红外线，或者只是纯信息例如金融数据、天气数据、网络信息、文本等。HTM 是可以被应用到许多领域的预测机器。

HTM 原理

在这部分我们会涉及到 HTM 的一些核心原理：为什么层级组织结构是重要的，HTM 的区域是如何构建的，为什么数据要以稀疏离散表征的形式储存，为什么基于时间的信息是关键。

层级性 一个 HTM 网络是由按层级排列的区域所组成的。区域是一个 HTM 网络的主要记忆和预测单元，我们将会在下面的部分详细介绍。具有代表性的是每个区域代表层级中的一个等级。层级会随着等级的上升不断聚合，下级多个子元素将被聚合到一个上级元素中，然而，由于反馈连接的存在，信息也会随着等级的下降不断分流。（层级和区域基本上是同义词。当描述区域内在功能的时候我们使用“区域”，当描述区域对于整个层级的影响时我们使用“层级”一词。

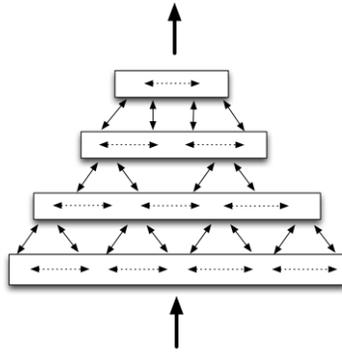


图 1.1：一个 4 区域在 4 层级中如何排列的简单示例，层内信息，层间信息，层级外部的输入输出

将多个 HTM 网络组合到一起时可能的。这种结构使得从多个数据源和感受器提供数据变得有意义。举个例子，一个网络可能处理音频信息另一个网络可能处理视觉信息。独立的网络之间存在聚合，并且独立的分支只向上聚合。

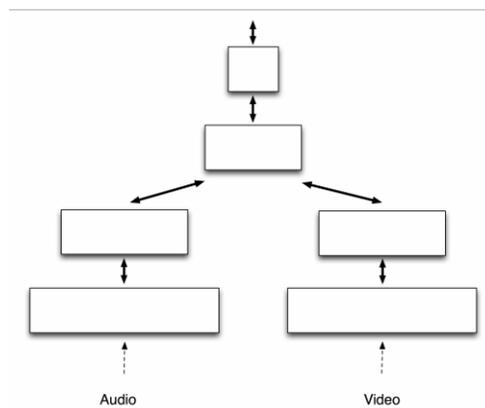


图 1.2：不同传感器网络的聚合

层级组织结构的好处就是高效率。它显著减少了训练时间和使用的记忆空间，因为每一层级中已学会的模式将以一种新奇的方式在更高的层级中得到重用。为了说明，让我们以视觉为例。在层级的最底层，你的大脑中储存着视觉区域中极小组成部分的信息，例如边和角。边是许多对象组成的基本构件。这些低级模式将会在中级层级中重组为更复杂的模式，例如曲线和条纹。一条弧线可以是耳朵的轮廓，方向盘的上部，咖啡杯的杯把。这些中级模式进一步组合成代表高级对象的模式，例如头、汽车、房子。学习这些新的模式你不需要重学它的构件。

另一个例子，想想当你学习一个新单词的时候，你不需要再学习字母、音节、音素。

层级中共享的表征也导致了预测能力的泛化。当你看见一种新的动物，如果你看到一张嘴和牙齿你会预测这个动物用他的嘴来进食并且可能会咬你。层级结构使得新对象通过它的子构件来继承它的已知属性。

HTM 层级中的一级能学习多少内容？或者换句话说，层级一共需要多少级？每级存储多少信息和一共需要多少级之间存在一个权衡问题。幸运的是，HTM 可以自动从输入和已分配资源的统计资料中得出最佳的表征分配方式。如果你给某级分配了较多的空间，那级将会构建较大、较复杂的表征，这也就意味着可能需要更少的层级。如果你分配较少的空间，该层级就会构建较小、较简单的表征，这也就意味着可能需要更多的层级。

至此为止我们已经讨论了许多较难的问题，例如视觉识别（“识别”指模式识别）。但是许多有价值的问题比视觉还要简单，并且一个 HTM 区域就足以解决。例如，我们用一个 HTM 网络来预测一个人浏览一个网站的时候下一步可能会点击哪里。这个问题涉及到使用网络的点击数据流去饲养（训练）HTM 网络。在这个问题中仅有一点或者没有空间上的层级性，解决方案主要需要研究临时的统计数据，也就是通过识别用户典型的操作模式来预测用户下一步点击那里。HTM 里的临时学习算法是这类问题的理想选择。

总而言之，层级能减少训练时间，减少使用的记忆空间，并且提供一种普适的框架。然而，许多简单预测问题只用一个层级就能解决。

区域

层级中的区域概念来自于生物学。新大脑皮层是一张约 2 毫米厚的巨大的神经细胞层。生物学家主要依据这些区域互相的连接方式来把新大脑皮层划分成不同“区域”。一些区域直接从感受器接受输入而另一些区域只接受传经几个区域的输入。层级性的定义就是这种从区域到区域的连通性。

所有新大脑皮层的区域在细节上看起来都是相似的。他们除了尺寸不同和在层级中位置不同之外都是相同的。如果你做个 2 毫米厚的新大脑皮层的区域横切片，你会看见 6 层，5 层细胞层和 1 层非细胞层（一般是这样当然也有例外）。新大脑皮层区域的每一层内有许多互相连

接的成列的细胞。HTM 区域也是由一层成列的高度内联的细胞组成的。新大脑皮层中的“第三层”是神经的主要前馈层之一。HTM 区域中的细胞略等于新大脑皮层“第三层”区域中的神经细胞。

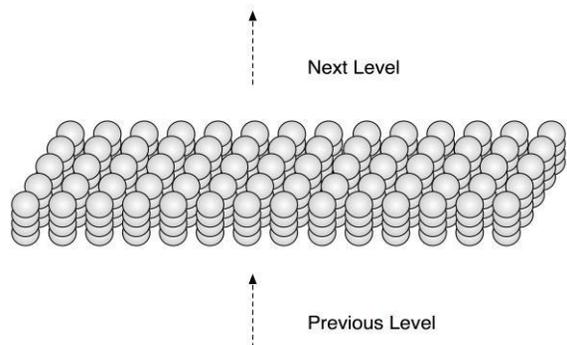


图 1.3 : HTM 区域的一部分。HTM 区域由许多细胞组成。细胞以成列的二维平面形式组织在一起。图中所示是一个每列 4 个细胞的 HTM 区域的一部分。每一列连接到另一子集的输入并且每个细胞和区域中的其他细胞相连（连接未画出）。注意 HTM 区域以及它的柱状结构等同于新大脑皮层区域中的一层神经细胞。

尽管 HTM 区域只在一部分上等同于新大脑皮层区域，它可以对复杂的数据流进行识别和预测，因此它可以解决许多问题。

稀疏离散表征

虽然新大脑皮层中的神经元高度连通，抑制神经元保证同一时间只有很少一部神经元处于活跃状态。因此，大脑中的信息总是通过巨大数量的神经细胞中的一小部分活跃细胞来表示。这种编码方式就叫做“稀疏离散表征”。“离散”意味着众多神经元的激活是为了表示一定的事物。一个神经元传达着一定的意义，但它必须被放到数量众多的神经元环境中解释才能传达完整的意义。

HTM 区域也使用稀疏离散表征。实际上 HTM 区域的记忆机制就依赖且仅依赖于稀疏离散表征。HTM 区域的输入总是离散表征，但它可能并不稀疏，所以 HTM 区域所做的第一件事就是将输入转化成稀疏离散表征。

举个例子，一个区域可能会接收到 20000 比特（位）的输入。输入位所含的“0”、“1”百分比会随着时间发生显著的变化。有时可能会含有 5000 个“1”，有时可能会有 9000 个“1”。HTM 区域可以将输入转换成一个 10000 位的，只有 2% 或者 200 位活跃（即发生变化）的内部表征，无论输入中有多少个“1”。随着时间的变化，HTM 区域的输入也会变化，内部表征也会随之改变，但是在 10000 位中总是只有 200 位是活跃的。

潜在输入模式的数量远大于区域中潜在表征的数量使得看起来像这一过程导致了大量的信息丢失。然而这两者的数量都是巨大的。一个区域的实际输入只是所有潜在输入的冰山一角。稍后我们将介绍怎样根据一个区域的输入创建一个稀疏表征。理论上信息的丢失不会产生实际影响。

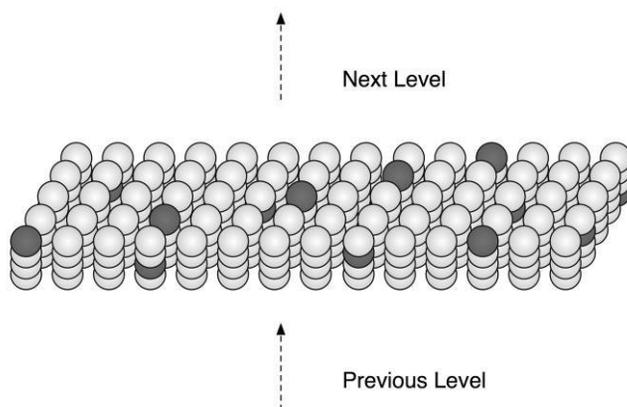


图 1.4：一个含有稀疏离散活跃细胞的 HTM 区域

稀疏离散表征有些令人满意的特性并且是 HTM 的重要组成部分，我们将在后面再次谈及。

时间的角色

时间在学习、识别和预测中扮演着举足轻重的角色。

让我们先从识别说起。如果没有时间，我们将几乎不能识别任何东西通过我们的触觉和听觉。举个例子，如果你眼睛被蒙上了，有人在你手上放了一个苹果，你可以通过触摸它识别出你手中的物体。当你的手指在苹果上移动时，景观触觉的信息在不断变化，但是对象本身（苹果），和你高级认知中“苹果”这一概念是保持恒定的。然而，如果把一个苹果放在你手掌上并且不允许你用手指去触摸，那么你将很难分辨出它是一个苹果还是一个柠檬。

对于听觉这一点同样适用。一个静态的声音只能传达很少的意义。一个单词，例如“苹果”，或者某人嚼碎苹果的声音，只能通过大量快速、连续的基于时间变化的音程来识别

视觉与之相比情况比较特殊。不像触觉和听觉，人类可以识别图像，当他们从眼前一闪而过以至于没有机会给你的眼球转动的时候也是如此。因此，视觉识别不是总是需要基于时间变化的输入。但是在一般情况下我们不断地在移动我们的眼睛、头、身体，而我们观察的对象也在我们周围不停地运动。我们视觉的快速识别能力是一种特殊情况，是长时间训练的结果。视觉、听觉、触觉在一般情况下都需要基于时间变化的输入。

在谈论了识别的一般情况和视觉识别静态图像的特殊情况后，让我们来看看学习。为了能够学习，所有的 HTM 系统在训练时必须接受基于时间变化的输入。即便是有时能够识别静态的视觉，也必须在学习对象外貌的时候看着变化的图像。举个例子，想象一下有一只狗正在朝你跑来。在任意时刻，这只狗在你的视网膜上形成了一个活跃的模式。你能够理解这些都是同一只狗在不同视角下的不同模式，但是从数学角度讲这些模式是截然不同的。大脑通过连续的观察从而认定这些不同的模式代表着同一只狗。时间在这里扮演着“监督”的角色（应该是“监督型学习算法”中的概念，译者注），告诉你将那些稀疏的模式组合在一起。

需要注意的是感官输入不仅仅是变化的就够了。一系列前后不相关的感官模式只会导致混乱。基于时间变化的输入必选来自同一个对象。仍须注意的是尽管我们用人类感官举例但一般还是多应用在非人类感官上。如果我们想训练 HTM 去识别来自一个动力装置的温度、振动、噪音传感器的模式，HTM 就需要来自这些传感器随时间变化的数据来训练。

一般来说，一个 HTM 网络需要大量的数据来训练。你学会如何识别狗是通过观察了许多品种的狗的实例，不是仅看了一只狗一眼。HTM 算法的工作是学习时间上连续的输入数据流，换句话说是一个能知道某些序列后面跟着哪些序列的模型。这项工作是困难的，因为我们可能不知道序列将从何时开始和结束，并且可能同时还重叠着其他的模式序列，学习就必须连续地进行，并且在有噪音的环境中进行。

学习和识别序列是能够预测的基础。当一个 HTM 知道一个模式可能跟在哪些模式后面时，他就可以通过当前输入预测下一个模式。稍后我们会涉及更多预测的细节。

我们现在回到 HTM 的四个基本功能：学习、识别、预测和行为（反应）。每一个 HTM 区域执行前三项功能：学习、识别和预测。然而行为是不同的。我们从生物学角度可知绝大多数新大脑皮层区域在形成行为中有着席之地，但是我们不认为对于许多有趣的应用行为是必不可少的。因此我们并没有将行为包含在当前 HTM 的实现中。我们只是为了完整性才在这里提到它。

学习

一个 HTM 区域通过发现模式并将它们从传感数据中序列化来了解世界。区域并不“知道”当前输入代表着什么。它处在一个纯粹的统计学领域内。它寻找着经常一起出现的输入位（bit）组合，也就是我们所说的空间模式。接着它研究这些空间模式怎样依次随着时间的变化而出现，也就是我们所说的实时模式或序列。

如果区域的输入来自一栋建筑周围环境的传感器，区域会发现建筑北侧的温度与湿度的组合不同于建筑南侧的。然后它会随着日积月累学会这些组合如何变化。

如果一个区域的输入代表着某家商店的销售，HTM 区域会发现某些类型的商品在周末销售，或者是当天气寒冷时的晚上某些价格幅度是受欢迎的。那么它就可以知道紧随在熟悉的序列化模式后出现的各种销售数据。

单个的 HTM 区域学习能力是有限的。一个区域会根据现有的记忆空间和输入数据的复杂性自动调整学习内容。如果分配给一个区域的记忆空间减少了，这个区域学习的空间模式也会根据需要简化。同样，当区域的记忆空间增加了，学习的空间模式也会相应地复杂化。如果一个区域医学会的空间模式是简单的，那么就需要一个层级结构去理解复杂的信息（前面所提到的每层的空间与层数是个权衡点，译者注）。我们可以从人类视觉系统中看出这一特性。新大脑皮层区域从视网膜接收到只表示视觉区域中很小一部分的空间模式。只有经过好几个层级的组合，空间模式才能表示整个或者绝大部分的视觉区域。

正如一个生物系统，HTM 区域中的学习算法能够“在线学习”，也就是说它可以通过最新的输入来学习。这里不需要从识别的阶段先划分出一个学习阶段，尽管通过附加的学习可以促进识别。随着输入模式的变化 HTM 区域也会逐渐发生变化。

经过初始化训练后，一个 HTM 可以继续学习或者在训练阶段完成后废除学习功能。另一种选择是只关闭最低层级的学习功能，在更高的层级依然继续学习。当一个 HTM 已经学会其世界的基本组成结构，更多的学习会发生在层级中的更高级内。如果 HTM 遇到以前没有见过的低层级结构，那么 HTM 将会用更长的时间去学习这些新模式。我们可以从人类身上发现这一特性。学习一个你已掌握的语言的新单词是相对容易的。然而，如果你尝试学习一个外语中有着不熟悉发音的单词，你会发现这会难很多，因为你不知道这些低层级的发音。

能够简易地发现模式是一种具有潜在价值的能力。能理解市场波动、疾病、天气、制造业中的高级模式，或者复杂系统中的风险，例如电力网，是极具价值的。即便如此，学习空间的实时的模式对识别和预测来说也是一个创举（意思是模式识别其实是个古老的问题，译者注）。

识别

当一个 HTM 学会了模式之后，它就可以对前所未见的输入进行识别。HTM 会在接收输入的时候与以前学过的空间实时模式进行匹配。新输入与旧序列的成功匹配是识别和模式匹配的关键。

想想你如何识别一段旋律。听到第一个音符之后你只能得到一点信息。第二个音符显著地缩小了可能性范围但是仍然不够。一般情况下你需要听到 3、4 个甚至多的音符才能认出一段旋律。HTM 区域的识别与之相似。它时刻注意着输入的数据流并将它们与之前学过的模式序列进行匹配。HTM 区域可以从一个序列的起始处发现匹配，但是多数情况下这个位置是不固定的，就像你从一段旋律任意处开始都可以识别出来。由于 HTM 区域使用离散表征，其对于序列记忆空间的使用和识别比旋律的例子要复杂的多，但是这个例子有助于我们认识它是如何工作的。

这可能不是显而易见的，但实际上你的感官经验对于你以前的经验来说都是新奇的，

即便你可以轻易地发现新奇输入中熟悉的模式。举个例子，你可以理解任何人说的“早餐”一词，不论他们是年长或是年轻，男性或是女性，语速快还是慢，或者是否带有很重的方言口音。甚至是你听同一人说“早餐”一百次，你的耳蜗（听觉感受器）也不可能受到两次完全相同听觉刺激。

HTM 区域和你的大脑面临同样的问题：输入可能永远不会重复。所以 HTM 区域和你的大脑一样必须想办法在识别和训练的时候处理新奇的输入。HTM 区域处理新奇输入的一种办法是通过使用稀疏离散表征。稀疏离散表征的一个关键特性就是你只用比较模式中的一部分就可以断定是否匹配。

预测

HTM 的每一个区域都储存着模式序列。通过当前输入和储存序列的比较，一个区域会形成对接下来输入可能是什么的预测。HTM 区域实际上储存的是稀疏离散表征之间的渐变。在一些实例中这种渐变可被看做是一种线性序列，就像一段旋律中的音符，但是在多数情况下会同时预测多个可能的输入。HTM 区域会依据可追溯到很久之前的上下文环境来产生不同的预测。HTM 中大多数的空间都被用来储存序列或者储存空间模式之间的渐变。

下面是 HTM 预测的一些关键特性。

1) 预测是连续的

你总是在不停的预测，即使你自己没有意识到。HTM 也是如此。当你在听歌的时候你会预测下一个音符。当你下楼梯的时候你会预测你的脚什么时候碰到下一级台阶。当看到投手掷球你会预测球将接近击球手。在 HTM 区域中，预测和识别基本上是一样的道理。预测不是独立的某一部分，而是贯穿于整个 HTM 区域的工作机制当中。

2) 预测发生在层级中的每一层

如果你有一个 HTM 区域的层级结构，那么预测会发生在每一层。区域会根据它们已掌握的模式进行预测。以语言为例。低级区域可能会预测下一个音素，而高级区域可能会预测下一个单词或词组。

3) 预测对上下文环境是敏感的

预测是基于过去所发生的事，并且和现在正在发生的也有关系。因此一个输入可能会由于先前环境的不同而产生不同的预测。一个 HTM 区域会根据需要去使用先前的环境，并且

上下文环境在时间上可长可短。这种能力被称为“可变顺序记忆”。举个例子，想象一下记忆像葛底斯堡演说这样的演讲。为了预测下一个词，只知道当前的词是远远不够的。在第一句中，单词“and”后面跟着“seven”并且后面又有一个“and”后面跟着“dedicated”。有时候只是一小段上下文就可以帮助我们预测；知道了“four score and”就可以帮我们预测出“seven”。而在另外一些时候会存在着重复的词语，并且可能需要一个大时间跨度的上下文才能知道你处于演讲中的什么位置，以及接下来是什么。

4) 预测是趋于稳定的

区域的输出就是预测。HTM 的特性之一就是区域的输出会变得稳定—变化的更慢，在更高层级驻留时间更长。这些特性是由区域预测的机制造成的。一个区域不是仅仅预测接下来会发生什么。如果它可以，它会提前预测很多步。我们假设一个区域可以提前预测五步。当一个新输入来临，最新预测的那步会改变但是之前预测的四步可能不会改变。结果就是，即便每个新输入都是完全不同的，由于只有一部分输出发生改变，所以使得输出相对于输入更稳定。这一特性反映了我们现实世界的经验，高级意识要比低级意识变化的慢得多，就像一首歌的名称比它实际的音符更难发生改变。

5) 预测可以告诉我们一个新输入是否是期望的

每个 HTM 区域都是一个新鲜事物检测器。因为每个区域都会预测接下来将发生什么，所以当有未预期的事发生的时候它会“察觉”。HTM 可以同时预测很多接下来可能的输入。所以它可能并不能准确预测接下来会发生什么，但是如果下面的输入不符合预测中的任何一个，HTM 区域会知道发生了异常。

6) 预测有助于增强系统的鲁棒性

当一个 HTM 预测出接下来会发生什么，预测会使系统识别的结果更倾向于它预测的结果。举个例子，当一个 HTM 处理自然语言的时候，它会预测接下来将发出怎样的声音、单词和观点。预测帮助系统补全缺失的信息。如果接收到一个有歧义的声音，HTM 会将它识别成它所预期的结果，这样便帮助了系统在噪音中识别。

序列记忆、识别、预测在一个 HTM 区域中是一个密不可分的整体。它们共同构成一个区域的核心功能。

行为

我们的行为会影响我们的感知。当我们转动眼球时，我们的视网膜会接收到变化的感官输入。移动我们的四肢和手指会将变化的触觉输入传输到大脑。几乎我们的一举一动都会影响到我们将感受什么。感官输入和运动行为是紧密联系在一起。

数十年来主流观点认为新大脑皮层中的一个区域，初级运动区，是运动命令在新大脑皮层中的发源地。随着时间的推移，人们发现新大脑皮层中的大多数区域都有运动输出，甚至是在低级感官区域。它使得整个皮质区域看上去是一个包含感官与运动功能的整体。

我们希望把运动输出添加到当前已实现框架的每个 HTM 区域中去，因为产生运动输出和产生预测有些相似。然而至今为止所有的 HTM 实现都是纯感官的，没有一个运动组件。

在实现 HTM 方面取得的进步

我们在将 HTM 理论框架转化成实际技术方面已经取得了实质性进展。我们已经实现并测试了几个版本的 HTM 脑皮质学习算法并且找到了可发布的基本架构。随着我们用新的数据集测试算法，我们会改善算法并且补全缺失的部分。我们会根据我们的工作更新这篇文档。下面三章描述了算法当前所处的状态。

还有很多理论中的部件没有被实现，包括注意力，区域间的反馈，特殊的计时（specific timing），以及行为/感官-运动机制。这些缺失的部件应该适用于已经建立的框架。

第二章：HTM 脑皮质学习算法

本章主要讲述在一个 HTM 区域中学习算法是如何工作的。第三、四章用伪代码描述了如何实现学习算法，而本章主要是一些概念。

术语

在我们开始之前，一个术语的注解可能是有帮助的。我们用神经系统科学的语言来描述 HTM 学习算法。像神经元、突触、潜在突触(potential synapses)、树突区域、柱状区域（新大脑皮层中）一类的术语将会经常出现。这些术语的使用是合理的，因为根据理论上的需要，本学习算法是通过比对大量神经系统科学事实而得出的结果。然而在实现算法的过程中我们不得不面对性能问题，因此每当我们觉得我们弄清了某些东西是怎么工作的时候我们会马上想办法提高处理速度。这样就使得有时候我们并不是严格遵循神经系统科学的有关事实，但我们得到了我们想要的结果。如果你对神经系统科学并不是很熟悉那么这不会是一个问题。但是如果你对神经系统科学的术语很熟悉，你可能会对我们使用术语与你理解的不同感到很迷惑。附录中生物学部分详细讨论了 HTM 学习算法与神经生物学中对应概念的相同点与不同点。下面我们会提到一些最可能造成你迷惑的一些概念偏差。

细胞的状态

HTM 的细胞有三种输出状态，由前馈输入激活，由横向输入激活（表示预测），以及非活跃状态。第一种输出状态相当于神经元中动作电位的一个短脉冲。第二种输出状态相当于神经元中较缓慢、较平稳的动作电位。我们认为没有必要为单独的动作电位建模或者是在这两种活跃状态之外再划分新的活跃等级。离散表征的使用貌似解决了对细胞内活跃等级分级的需要。

树突区域

HTM 细胞有相当真实的（因此也是复杂的）树突模型。根据理论每个 HTM 细胞有一个接近细胞核的树突和许多末端树突。靠近细胞中心的树突接收前馈输入，末端突触接收附近细胞的横向输入。有一类抑制神经元制约着整个柱状区域内的细胞对相似前馈输入的响应。为了简化，我们用一个柱状区域共享的树突来代替每个细胞上的近核树突。空间沉积池功能（稍后介绍）在柱状区域的水平上操作共享的树突。时间沉积池功能在柱状区域内每个细胞的水平上

操作末端树突。这种简化实现了与原来相同的功能，尽管在生物学上并不存在一个附属于柱状

区域的树突。

突触

HTM 突触有二值权 (binary weights)。生物学中的突触有着变化权，但是它们是部分随机变化的，这也就说明了生物学上的神经元不可能依赖确定的突触权。离散表征及树突操作模型使得我们可以在无副作用的情况下对 HTM 突触进行二值性赋值。为了对已形成和未形成的突触建模，我们引用两个你可能不是很熟悉的神经系统科学概念。一个概念是“潜在突触”。它是指所有足够接近树突的轴突可能会形成突触。第二个叫“连通值” (permanence)。这是对每个潜在突触设置的标量。一个突触的连通值是指树突与轴突之间连通性的程度。从生物学角度讲，这一程度由完全不连接开始，到开始形成一个突触但还未连通，再到满足最低的连通要求，最后到满足最大连通。连通值是一个关于突触的标量，值域为 0.0 到 1.0。学习的过程中包含了增加或减少突触的连通值。当一个突触的连通值超过阈值时，它就是连通的并且权值为“1”。当一个突触的连通值低于阈值时，它就是不连通的并且权值为“0”。

综述

想象一下你是一个 HTM 区域。你的输入由成千上万的二进制位组成。这些输入位可能代表着感官的数据或者来自层级中较低的另一区域。它们以一种复杂的方式影响着细胞。你会如何操作这些输入？

我们应经讨论过最简单的答案。每个 HTM 区域寻找输入中共有的模式并且学习这些模式序列。每个区域根据对序列的记忆来产生预测。这种高度概括的描述听起来很简单，但实际上发生事情远不止这些。让我们从以下三点更深入地探讨：

1. 构建输入的稀疏离散表征
2. 在有先前输入的环境中构建输入的代表
3. 在有先前输入的环境中根据当前输入构建预测

我们会更详细地讨论这三步。

1) 构建输入的稀疏离散表征

当你想象一个区域中的输入的时候，你可以认为是数量巨大的二进制位。在大脑中这些就是神经元延伸出的轴突。在任意时间点这些输入的其中一部分是活跃的（值为 1）另一些是不活跃的（值为 0）。输入的活跃百分比是不断变化的，可以说在 0%到 60%之间。HTM 区域所做的第一件事就是把输入转换成新的稀疏的表征。举个例子，输入可能有 40%是“1”但是新的表征可能只有 2%是“1”。

一个 HTM 区域从逻辑上讲是柱状区域的集合。一个柱状区域又是由一个或多个细胞组成。柱状区域在逻辑上可能是按二维方式排列，但这不是必要条件。每个柱状区域都关联着输入中一个唯一的子集（通常会与其它柱状区域重叠但不会完全相同）。结果就是不同的输入模式导致了柱状区域活化程度的不同。有很强抑制作用的柱状区域相对来说活化程度较低。（抑制通常发生在一个半径范围内，它可以从某些局部扩展至整个区域。）输入的稀疏表征就是对哪些柱状区域活跃，哪些在抑制后不活跃的一个编码。抑制机制主要是为了实现将柱状区域的活跃百分比控制在一个相对稳定的范围内，即便是有大量变化显著的输入的时候。

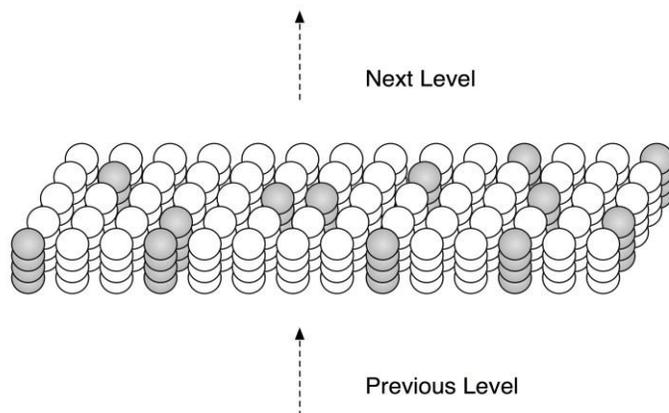


图 2.1: 一个 HTM 区域包含着由细胞组成的柱状区域。这里只是区域中的一小部分。每个柱状区域的细胞从输入的唯一对应子集收到激活信号。有强烈活性抑制的柱状区域活化程度较低。结果就产生了输入的稀疏离散表征。图中用灰色表示活跃的柱状区域。（在没有先前状态的情况下，所有的细胞都是活跃的。）

现在想象一下输入模式发生了改变。如果只发生了很小的改变，有些柱状区域会接收到多一点或者少一点的“1”，但是整个活跃柱状区域的集合可能不会发生太大改变。因此，相似的输入（“1”的个数相差不多）会映射到一个相对稳定的活跃柱状区域集合。编码的稳定

很大程度上依赖于柱状区域所关联的输入。这种关联将通过一种方法来建立，稍后会介绍。

这些所有的步骤（建立输入子集到每个柱状区域的连接，决定每个柱状区域的输入水平，使用抑制来筛选一个稀疏的活跃柱状区域集合）被称为“空间沉积池”。这个术语是指在“空间”上相似（指有着数量众多的相同的“1”）的模式被沉积（指它们组合成一个共同的表征）。

2) 在有先前输入的环境中构建输入的表征

区域执行的下一个功能是把柱状区域的表征转换成一种新的包含过去状态或环境的表征。新表征通过激活每个柱状区域中的一个细胞子集而构成，对 HTM 而言通常每个柱状区域只有一个细胞参与（图 2.2）。

思考一下当你听到两句话，“I ate a pear”和“I have eight pears”。单词“ate”和“eight”是同音异义词，它们发音完全相同。我们可以确信大脑中某处的神经元对“ate”和“eight”都做出了反应。毕竟不可区分的声音进入了耳朵。然而我们也可以确信在大脑另一处的神经元对在不同上下文中的输入做出不同响应。在你听到“I ate”时，声音“ate”的表征是不同于当你听到“I have eight”的。想象一下你已经记住了“I ate a pear”和“I have eight pears”。当你听到“I ate ...”时会产生一个和听到“I have eight...”不同的预测。因此在听到“I ate”和“I have eight”后一定有不同的内部表征。

对不同环境下的输入进行不同的编码，这一原则是感知和运动的普遍特点，并且也是 HTM 区域最重要的功能之一。对于这一能力，再怎么重复强调其重要性也不为过。

HTM 区域中的每个柱状区域由许多细胞组成。同一柱状区域的细胞有相同的前馈输入。柱状区域的每个细胞可以是活跃的或非活跃的。通过在活跃的柱状区域中选择活跃的细胞，我们可以将不同环境中完全相同的输入加以区分地表示出来。一个特例可能会有所帮助。假设每个柱状区域有 4 个细胞并且每个输入的表征由 100 个活跃的柱状区域构成。如果同一时间同一柱状区域内只有 1 个细胞是活跃的，我们就有 4^{100} （4 的 100 次方）种表示完全相同输入的方法。相同的输入总是会导致相同的 100 个柱状区域兴奋，但是在不同的环境中这些柱状区域活跃的细胞是不同的。现在我们可以表示在各种环境中的相同输入，但怎样使那些不同的表征变的唯一呢？基本上全部随机地从 4^{100} 个可能的模式中选取一对，大概会有 25 个细胞是重复的（即同一柱状区域内活跃细胞相同）。因此，一个特定输入在不同环境中的两个表征会有大约 25 个共同的细胞和 75 个不同的细胞，这使得它们很容易被区分。

下面是 HTM 区域使用的通用规则。当一个柱状区域被激活时，它会关注柱状区域内所有细胞。如果柱状区域中有一个或多个细胞已经处于预测状态，那么只有这些细胞会被激活。如果柱状区域中没有处于预测状态的细胞，那么所有的细胞都会被激活。你可以这样想，如果一个输入模式是预期中的，那么系统会通过只激活处于预测状态的细胞来进一步确认。如果输入模式不是预期中的，那么系统会激活所有柱状区域内的细胞，就好像在说“输入是意料之外的所以所有可能的解释都是有效地”。

如果没有前置状态，并且因此没有上下文环境和预测，柱状区域中的所有细胞会随着柱状区域的激活而激活。这种情况与听到歌曲的第一个音符时相似。没有了上下文环境你通常无法预测下面会发生什么，所有的选项都是有可能的。如果有前置状态但是输入与预期不相符，那么被激活的柱状区域中所有的细胞也将被激活。这个决断是由柱状区域依据其根本原理而下达的，因此符合预测或不符合预测两者必有一个且仅有一个发生。

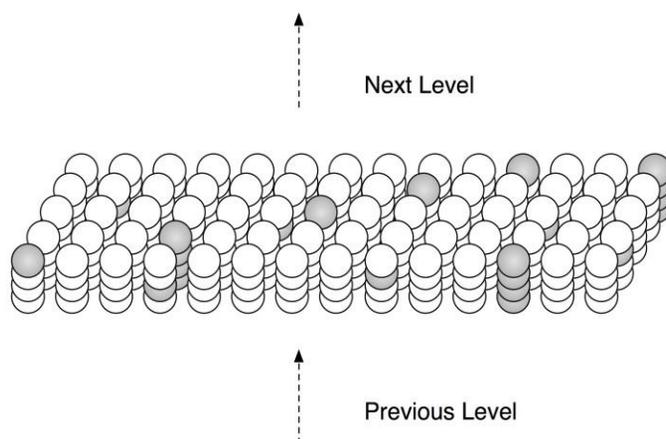


图 2.2：通过激活每个柱状区域的细胞子集，一个 HTM 区域可以表示不同环境下的相同输入。柱状区域只激活有预测的细胞。没有预测细胞的柱状区域激活所有细胞。图中显示了有些柱状区域是部分细胞活跃，有些是全部细胞活跃。

正如之前术语部分所提到的，HTM 细胞可以是三种状态之一。如果一个细胞被前馈输入激活，我们用“活跃”一词。如果细胞被横向的附近其它细胞激活，我们说它处于“预测状态”（图 2.3）

3) 在有先前输入的环境中根据当前输入构建预测

我们的区域所做的最后一步就是做出一个接下来可能会发生什么的预测。预测是基于第二步中生成的表征的，其包含了先前输入所构成的环境。

当一个区域产生一个预测的时候，它会激活（进入预测状态）所有可能因将来的前馈输入而变得活跃的细胞。由于区域内的表征是稀疏的，所以可以在同一时间进行多种预测。举个例子，如果有 2% 的柱状区域因一个输入而变得活跃，你可以认为 10 个不同的预测会导致 20% 的柱状区域有 1 个处于预测状态的细胞。或者有 20 个不同的预测会导致 40% 的柱状区域有 1 个处于预测状态的细胞。如果一个柱状区域有 4 个细胞，随着一次激励的发生，10% 的细胞会处于预测状态（ $40\% \times 1/4$ ）。

后面关于稀疏离散表征的章节会介绍，即便不同的预测混合在一起，区域也可以高度确定一个特定输入是否产生了预测。

一个区域怎样做出预测？当输入模式随着时间变化时，不同的柱状区域以及细胞的集合按一定的顺序变得活跃。当一个细胞变得活跃时，它会优先与附近马上能激活的细胞子集建立连接。这种连接建立的快慢是由应用程序的学习速率决定的。然后一个细胞所需做的就是注意这些连接是否有激励。当有激励从连接传来时，细胞会意识到它将要被激活并进入预测状态。因此一个细胞集合的前馈激励通常会导致另一个细胞集合的预测激励。想想当你认出一首歌并开始预测下一个音符的时候。

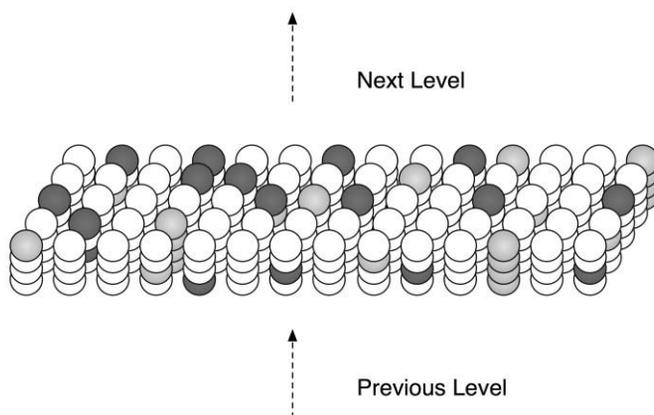


图 2.3：在任意时刻，HTM 区域中的一些细胞会被前馈输入所激活（图中浅灰色）。收到活跃细胞的横向输入的细胞会进入预测状态（图中深灰色）

总之，当一个新输入会导致一个稀疏的柱状区域集合活跃。柱状区域中的一个或多个细胞将变得活跃。这种改变通过经学习建立的细胞间的连接使其它细胞进入预测状态。经区域内连接激活的细胞（处于预测状态）构成了接下来可能发生什么的预测。当下一个前馈输入来临时，它

又会选择另一个稀疏的柱状区域集合。如果一个非预期的柱状区域（没有预测状态的细胞）被激活，也就是说它不在任何细胞的预测之中，它会激活该柱状区域内所有的细胞。如果一个被激活的柱状区域有 1 个或多个处于预测状态的细胞，那么只有这些细胞会被激活。区域的输出就是所有细胞的活跃状态，包括因前馈输入而活跃的细胞和处于预测状态的细胞。

正如之前所提到的，预测不是仅仅预测下一步。HTM 区域是对未来好几步进行预测。还是用旋律的例子，HTM 区域不只是预测旋律的下一个音符，它可能会预测下面四个音符。这便造就了一个令人满意的特性。一个区域的输出（区域中所有处于活跃及预测状态的细胞集合）比输入变化的要慢。想象一下区域正在预测一段旋律中的下面四个音符。我们用字母序列“ABCDEFGG”表示旋律。在听到前两个音符后，区域识别出了序列并开始预测。它预测出了“CDEF”。“B”细胞已经处于活跃状态，所以表示“BCDEF”的细胞必属于两种活跃状态之一。现在区域听到下一个音符是“C”。处于活跃及预测状态的细胞集合现在变成“CDEFGG”。需要注意的是输入模式由“B”到“C”被完全改变，但是只有 20%的细胞发生了变化。

因为 HTM 区域的输出时表示细胞活跃情况的矢量，所以输出在本例中比输入要稳定 5 倍。在层级区域中，我们会发现随着层级的升高，实时稳定性也会随之增长。

我们用术语“时间沉积池”来表示为表征添加上下文环境和预测这两步骤。通过对模式序列构建变化缓慢的输出，我们实际上将时间上连续的不同模式“沉积”在一起。

现在我们将更深入具体细节。我们将从时间沉积池和空间沉积池所共有的概念讲起。然后我们会讨论空间沉积池所特有的概念和细节，最后是时间沉积池所特有的概念和细节

共有的概念

时间沉积池和空间沉积池中的学习是相似的。两种学习都包括在细胞间建立连接，或者说突触。时间沉积池负责同一区域内细胞间的连接。空间沉积池负责输入与柱状区域之间的前馈连接。

二值权

HTM 突触只有 0 或 1 两种结果。他们的“权值”是二元的，这一特性不同于其它的神经网络模型用 0 到 1 的标量变量来表示。

连通值

突触在学习过程中不停的形成和分解。正如之前所提到的，我们为每个突触设置一个标量（0.0 到 1.0）来反映连接的连通性。当一个连接得到加强时，它的连通值会增加。在相反的环境下连通值会减少。当连通值高于阈值（比如 0.2）时，可以认为突触已建立。如果连通值低于阈值则突触失效。

树突区域

突触连接到树突区域。有两种树突区域，靠近细胞体的和处于末端的。

-靠近细胞体的树突与前馈输入构成突触。这种活跃的突触线性加和可以决定柱状区域的前馈激活。

-末端树突区域与同区域内的其它细胞构成突出。每个细胞有多个末端树突区域。如果一个末端树突区域的突触活跃度之和超过了阈值，那么该细胞将会进入预测状态。由于每个细胞存在多个末端树突区域，所以细胞进入预测状态是几个胞内阈值探测器进行逻辑“或”操作的结果。

潜在突触

之前曾提到过，每个树突区域有一份潜在突触的列表。所有的潜在突触被赋予了一个连通值并且在该值超过阈值时变成起作用的突触。

学习

学习过程涉及树突区域的潜在突触的连通值的增加和减少。突触连通性增减的规则与“赫布”（Hebbian）定律相似。举个例子，如果一个突触后的细胞因树突区域接收到的输入超过阈值而激活，那么这个树突区域中突触的连通值将被修改。活跃的突触，并且因此而有助于细胞被激活的，它们的连通性会增加。不活跃的突触，并且因此而对活跃无所贡献的，它们的连通性会下降。在时间沉积池和空间沉积池中具体更新哪些突触的连通值的情况是有所不同的。我们会在下面详细讨论。

现在我们要讨论针对时间沉积池和空间沉积池的功能性概念。

空间沉积池概念

空间沉积池的主要功能是将一个区域的输入转换成稀疏模式。这一功能是十分重要的，因为在学习序列和做出预测的机制中要求从稀疏离散表征开始。

空间沉积池有一些决定其如何操作和学习的目标是有重复部分的。

1) 使用所有的柱状区域

一个 HTM 拥有固定数量的柱状区域用来学习如何表示输入中的共有模式。有一个原则就是保证所有的柱状区域都学着去表示些有用的事物，无论有多少柱状区域。我们不希望有柱状区域从未活跃过。为了防止此类情况发生，我们记录一个柱状区域相对于相邻的区域的活跃频率。如果一个柱状区域的相对活跃度过低，它会增强它的输入活跃等级直到它变得活跃。事实上所有的柱状区域都与相邻区域竞争着成为表示输入的参与者。如果一个柱状区域不是很活跃，它会变得更具“侵略性”。此时，其它柱状区域会被迫修改它们的输入并开始表示有细微不同的输入模式。

2) 维持一定的密度

一个区域需要对它的输入构建一个稀疏表征。柱状区域的大多数输入会抑制周围其它的柱状区域。而抑制半径与柱状区域的感受域的大小成比例（因此可以从一小部分扩展至整个区域）。正因有抑制半径，我们只允许一定百分比的柱状区域在大多数输入中成为“赢家”。其余的柱状区域不会活跃。（抑制“半径”暗示了柱状区域是按二维排列的，但是这一概念同样适用于其它的拓扑结构。）

2) 避免琐碎模式

我们希望所有的柱状区域都表示输入中较重要的模式。这一目标可以通过设置激活柱状区域的输入的最小阈值来实现。例如，如果我们将阈值设置为 50，那就意味着一个柱状区域的树突区域中至少要有 50 个突触被激活，这样才能保证它所表示的模式有一定程度的复杂性。

3) 避免多余的链接

如果我们不注意，一个柱状区域会构成数量巨大的有效突触。这样会导致它对许多不相关的输入模式都有强烈的响应。不同突触子集会对不同的模式做出响应。为了避免这一问题，我们

减少对柱状区域当前活跃无贡献的突触的连通值。通过对无贡献的突触进行有效惩罚，我们可以保证一个柱状区域表示有限数量的输入模式，有时只有一个。

4) 自适应的感受域

真正的大脑是高度可塑的。新大脑皮层的区域可以学会表示完全不同的事物并对各种变化做出反应。如果新大脑皮层中的一部分被损坏，其它部分会调整自身并表示损坏部分所表示的事物。如果一个感受器官损坏或发生改变，新大脑皮层中与之相关联的部分会表示其它的事物（即放弃原来所表示的）。这一系统称为自适应。

我们希望 HTM 区域也能展现出同样的灵活性。如果我们为一个区域分配 10000 个柱状区域，它应该学会怎样才能最合理的用 10000 个柱状区域来表示输入。如果我们分配 20000 个柱状区域，它也应该学会怎么去合理利用。如果输入的统计信息发生改变，柱状区域也应该改变以便反应最真实的情况。总而言之，HTM 的设计者能够对区域任意分配资源，并且区域将会在可用的柱状区域和输入统计数据的基础上尽其所能地最优地表示输入。有一个通则，区域中柱状区域越多，每个柱状区域就会表示更大更多的输入模式的细节。一般来说柱状区域也会因此减少被激活的次数，但我们会维持一个相对稳定的稀疏水平。

实现这一令人高度满意的目标不需要新的学习法则。通过促进不活跃的柱状区域，抑制周围的柱状区域来维持稳定的稀疏度，确立输入的最小阈值，维持一个庞大的潜在突触池，并且基于突触的贡献来进行增减，所有的柱状区域会动态地进行设置以达到预期的效果。

空间沉积池细节

现在我们可以详细讨论空间沉积池具体做了些什么。

- 1) 一切始于一个由固定数量的二进制位组成的输入。这个输入可能表示感受器的数据或者来自于层级中较低的区域。
- 2) 为区域分配固定数量的柱状区域来接收输入。每个柱状区域有一个相关联的树突区域。每个树突区域有一个潜在突触的集合来表示输入的子集。每个潜在突触有一个连通值。基于它们的连通值，一些潜在突触会变得有效。
- 3) 对于给定的输入，柱状区域有多少有效的突触是与活跃输入位（即值为 1）有关的。

4) 活跃突触的数量要乘以一个“促进”系数，它是由一个柱状区域相对于周围柱状区域的活跃频率动态决定的。

5) 在受到促进后达到最高程度活跃的柱状区域会使在抑制半径内除固定百分比的柱状区域外的其它柱状区域暂时丧失机能。抑制半径是由输入的传播范围（或者称为“扇出”）动态决定的。这样我们便有了一个活跃柱状区域的稀疏集合。

6) 对于每个活跃的柱状区域，我们调整所有潜在突触的连通值。与活跃输入位相对应的突触的连通值会增加。与不活跃输入位对应的突触的连通值会减少。连通值的变化可能会使一些突触从有效变得无效，反之亦然。

时间沉积池概念

回忆一下时间沉积池怎样学习序列和做出预测。基本的方法是当一个细胞变得活跃后，它会与比它先活跃起来的细胞建立连接。然后细胞可以通过关注它们的连接来预测它们何时会被激活。如果所有的细胞都是如此，那么它们就可以储存和回忆起序列，并且它们可以预测接下来可能会发生什么。这里没有所谓的模式序列的储存中枢，取而代之的是记忆离散地存储在单个细胞内。正因记忆是离散的，所以系统有良好的鲁棒性。单个细胞会导致失败，通常对识别只起到一丁点或者没有作用。

值得注意的是时间沉积池所利用的几个稀疏离散表征的重要特性。

假设我们有一个区域总是用总的 10000 个细胞中的 200 个活跃细胞（任何时刻活跃细胞占 2%）来构建表征。我们怎样记忆和识别一个 200 个活跃细胞的特定模式。一个简单的做法是为我们关心的这 200 个细胞建立一个表。如果我们再次看见相同的 200 个细胞，那么我们就识别除了序列。但是如果只记录 200 个活跃细胞中的 20 个而忽略其余的 180 呢？会怎么样呢？你可能会认为只记录 20 个细胞会造成许多错误，那 20 个细胞会出现在许多不同的模式中。但这不是关键。因为模式是巨大并且稀疏的（在本例中是 10000 个细胞中的 200 个活跃细胞），记录 20 个细胞几乎和记录所有的细胞效果相同。实际系统中的出错率是极其小的并且我们也尽可能地减少使用的内存。

HTM 区域中的细胞利用了这一特性。每个细胞的树突区域都有一个到其它细胞连接的集合。

树突区域构建这些连接作为识别任意时刻网络状态的手段。在其附近可能有成百上千的活跃细胞但是树突区域只连接其中的 15 到 20 个。当树突区域发现其中有 15 个细胞活跃，它会相当确信较大的模式出现了。这一技术被称为“子抽样”并且其应用贯穿整个 HTM 算法。

每个细胞都参与了许多不同的离散模式和许多不同的序列。一个特定的细胞可能几十或几百个临时渐变的一部分。因此每个细胞有多个树突区域，不止一个。理想中的是每个细胞为每个它希望识别的活跃模式分配一个树突区域。但是实际上，一个树突区域可以学习几个完全不同的模式的连接并且很好地工作。例如，一个树突区域可能从 4 个不同的模式每个学习到 20 个连接，总过也就是 80 个连接。然后我们设定一个阈值，当有任意 15 个连接被激活那么树突区域就被激活。让我们来说说发生错误的可能性。这是可能的，在一定巧合下不同模式混合的 15 个活跃连接会达到树突区域的阈值。然而这种错误发生的概率很小，再说一次，这是因为表征的稀疏性。

现在我们知道了一个有十几、二十个树突区域和几千个突触的细胞是怎么是怎么识别成百的独立的细胞活跃状态的。

时间沉积池细节

这里我们列举一下时间沉积池的执行步骤。我们接着空间沉积池停止的地方，从表示前馈输入的活跃柱状区域集合说起。

- 1) 对于每个活跃的柱状区域来说，查找柱状区域中处于预测状态的细胞并将其激活。如果没有细胞处于预测状态，那么就激活全部的细胞。结果得到的活跃细胞集合是当前输入在先前的输入环境下的表征。
- 2) 对于区域中每个细胞的每个树突区域来说，统计有多少已建立的突触连接到了活跃的细胞。如果数量超过了阈值，那么该树突区域就会被标记为活跃。有活跃树突区域的细胞被置为预测状态，除非该细胞已经被前馈输入激活。没有活跃树突区域并且没有因自下而上输入而活跃的细胞将变得不活跃或者维持不活跃。此时区域中处于预测状态的细胞集合就是区域的预测。
- 3) 当一个树突区域变得活跃时，修改与之相关的所有突触的连通值。对于活跃树突区域中的每个潜在的突触，增加那些与活跃细胞相连接的突触的连通值并减少那些与不活跃细胞相连的

突触的连通值。突触连通值的这一变化被标记为临时的。

修改这些已经经过充分训练的树突区域的突触是为了更好地激活树突区域并产生出预测。然而我们总是想尽可能地预测得更远。因此，我们在同一细胞内再选取一个树突区域来训练。第二个树突区域我们选择和系统前几步的预测描述最匹配的树突区域。对于这个树突区域，通过使用系统前几步的预测描述，增加那些连接到活跃细胞的突触的连通值并且减少那些连接到不活跃细胞的突触的连接值。这种对突触连通值的改变被标记为临时的。

- 4) 当一个细胞因前馈输入而从不活跃变得活跃，我们会遍历每个与该细胞相连的潜在突触并且去掉一切临时性标记。因为我们只有当突触准确预测了前馈输入的活跃细胞后才会更新它的连通值。
- 5) 当一个细胞从活跃状态变得不活跃的时候，撤销该细胞的潜在突触中标记为临时的连通值变化。我们不希望强化没有正确预测前馈细胞活跃的突触的连通值。

注意只有细胞被前馈输入激活的细胞才会在区域内传播激励，否则预测会产生更远的预测（即将预测作为输入再进行预测）。但是所有的活跃细胞（包括前馈激活的和处于预测状态的）构成了区域的输出并且传播到层级中的更高层。

单一源与可变源的序列与预测

在我们结束有关时间沉积池和空间沉积池的讨论之前还有一个重要的话题要讨论。可能不是所有的读者都对此感兴趣，并且对于理解第三、四章这也不是必须的。

柱状区域有多个细胞的作用是什么？特别是当柱状区域只有一个细胞的时候会发生什么？

在前面的例子中我们说过一个包含 100 个活跃柱状区域的表征，每个柱状区域有 4 个细胞，一共有 4^{100} 种编码。因此同样的输入在不同的环境中也不会产生混淆。例如，如果输入模式代表单词，那么区域可以记住许多重复使用同一单词的句子并且不混淆。就像“dog”这个单词，在不同的环境中有不同的表征。这一能力允许一个 HTM 区域进行一种“可变源”预测（指在多种环境状态下进行预测）。

可变源预测不仅基于当前发生了什么，还基于大量变化的之前的环境。一个 HTM 区域的记忆是可变源的。

如果我们将每个柱状区域的细胞增加到 5 个，那么在我们的那个例子中对于特定输入的可用的编码数量就会增加到 5^{100} 个，相对于 4^{100} 增加了不少。但是这些数字太大了以至于对于许多实际问题仅是容量上的增加并没有什么用处。

然而，减少每个柱状区域的细胞数量却意义重大。

如果我们让每个柱状区域都只有 1 个细胞，那么我们将会失去在表征中包含环境的能力。一个区域的相同输入总是会产生相同的预测，并且无视先前的活跃情况。若每个柱状区域只有 1 个细胞，那么 HTM 区域的记忆就是“单一源”的，预测也只基于当前的输入。

单一源预测非常适合我们大脑所解决的一种问题类型：静态空间识别。正如之前所提到过的，人类可以识别出短暂视觉图像中的物体，即使短暂到眼球来不及转动。对于听觉来说，你往往要听一段模式序列才能识别出你听到的是什么。视觉通常也是如此，你通常处理的是一个视觉图像流。但是在某些环境下你可以在一瞬间识别出图像。

实时和静态识别看起来需要不同的识别机制。一个需要识别模式序列并且根据长度变化的环境做出预测。另一个需要在不借助实时环境的情况下识别静态的空间模式。每个柱状区域有多个细胞的 HTM 区域非常适合识别时间上连续的序列，每个柱状区域只有 1 个细胞的 HTM 区域适合识别空间模式。在 Numenta，我们进行了大量实验，将每个柱状区域只有 1 个细胞的 HTM 区域应用在解决视觉问题上。这些实验的细节超出了本章的范围，但是我们会讨论几个重要的概念。

如果我们把图像输入到一个 HTM 区域，那么柱状区域会学习如何表示像素的常见空间排列。学习的模式种类与新大脑皮层 V1 区域（新大脑皮层中的一个区域在生物学中被广泛研究）中所观测到的相似，诸如不同朝向的线条和角。通过用运动的图像训练，HTM 区域学会了这些基本形状是如何渐变的。例如某一位置的垂直线经常会有另一条垂直线左右移动。所有经常被观测到的模式的渐变会被 HTM 区域记住。

现在如果我们将一条向右移动中的垂直线的图像输入到区域会怎样呢？如果我们的区域每个柱状区域只有一个细胞，它会预测垂直线接下来可能出现在左边或者右边。它无法通过先前的环境知道垂直线过去的位置，因此也无法判断它是朝左运动还是朝右。你会发现这

些每个柱状区域只有一个的细胞从行为上看像新大脑皮层中的“复杂细胞”。这种细胞的预测性输出会成为在不同位置的可见直线的激励，无论这条直线是朝左移动还是朝右移动或者完全不动。我们进一步观测到这种区域在保持区别不同图像的能力的同时还展现出对平移、缩放等变化的稳定性。这种反应正是空间不变性（识别一个图像中不同位置的同一模式）所需的。

如果现在我们对多细胞柱状区域的 HTM 区域进行相同实验，我们会发现细胞的行为像新大脑皮层中的“调谐定向复杂细胞”。细胞的预测一定是正在向右或向左运动的直线，不可能是其它情况。

将这些结果综合起来我们得到以下假设。新大脑皮层一定进行单一源和可变源的识别与预测。新大脑皮层中的每个区域有四、五层细胞。每层细胞在几个方面是不同的，但他们共享柱状区域的响应内容以及层内巨大的水平连通性。我们推测新大脑皮层中的每一层细胞都遵循着类似于本章中描述的 HTM 识别和学习规则。不同层的细胞扮演着不同的角色。例如在解剖学中皆知的第六层建立层级间的反馈以及第五层与运动行为有关。第四层和第三层是两个初级前馈层。我们推断第四层和第三层的一个不同之处在于第四层的活动是独立的，也就是像单细胞柱状区域，与之相反，第三层的行为像多细胞柱状区域。因此新大脑皮层的区域对于感官输入有单一源和可变源记忆。单一源序列记忆（与第四层神经元基本相同）用来构建空间性不变化的表征。可变源序列记忆（与第三层神经元基本相同）用来识别和预测运动的图像。

总而言之，我们假定本章中描述的算法在新大脑皮层中的所有层都是适用的。新大脑皮层中的层在重要的细节方面存在不同，这也就使得他们扮演着与前馈和反馈、注意力、运动行为相关的不同角色。在靠近感官输入的区域，有一个执行单一源记忆的细胞层是很用的，这会导致空间不变性。

在 Numenta，我们对单一源（单细胞柱状区域）HTM 区域解决图像识别问题进行了实验。我们也对可变源（多细胞柱状区域）HTM 区域识别及预测可变源序列进行了实验。在将来，把这些集成到一个区域并为了其它目的扩展算法是合乎逻辑的事。然而我们相信许多有趣的问题都可以被单层、多细胞柱状区域的区域等价物所解决，无论是单独的一个区域还是处于一个层级结构之内。

第三章：空间沉积的实现及伪代码

本章包含了一个简单实现的空间沉积池功能的详细伪代码。本代码的输入是一大批自底向上的来自感受器或者低级区域的二进制输入。代码中计算的 *activeColumns(t)*—在 t 时刻因自底向上输入而活跃的柱状区域的列表。这个列表然后作为输入被送到下一章描述的时间沉积池程序，也就是说 *activeColumns(t)* 是空间沉积程序的输出。

伪代码按照发生的顺序被分为三个阶段：

阶段一：根据当前输入计算柱状区域的覆盖情况

阶段二：在抑制作用完成后计算最终活跃的柱状区域

阶段三：更新突触的连通值和内部变量

虽然空间沉积池的学习天生就是在线式的，但你可以通过跳过第三阶段关闭学习功能。

本章的剩余部分包含了这三个阶段伪代码。变量的数据结构和代码中用到的支持程序在最后有定义。

初始化

在接收任何输入之前，区域通过计算一份每个柱状区域的初始潜在突触列表来初始化。这由输入空间中选取的一个随机子集组成。每个输入用一个突触来表示并指定一个随机的连通值。随机连通值的选取有两条标准。第一，取值要在 *connectedPerm*（被认为“连通”的最小连通值）周围的一个小范围内。这能够使潜在突触在少量的迭代训练后处于连通（或不连通）的状态。第二，每个柱状区域在输入区域上有个天生的中心，并且连通值对中心存在偏袒（越靠近中心连通值越高）。

阶段一：覆盖

给定一个输入向量，第一阶段就是根据这个向量计算每个柱状区域的覆盖情况。柱状区域的覆盖情况可以简单地理解为与当前激励输入相连的突触数量，然后乘以它的促进系数。如果这个值小于 *minOverlap*，那么我们将覆盖值 (*overlap(c)*) 置为 0。

```
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)
```

阶段二：抑制

第二阶段计算哪些柱状区域在经过抑制作用后作为胜利者留下来。*desiredLocalActivity* 是用来控制最终活跃的柱状区域数量的参数。举个例子，假设 *desiredLocalActivity* 是 10，那么一个柱状区域如果想成为胜利者，它的覆盖值就得是它所在抑制半径内前 10 大的。

```
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.
```

阶段三：学习

第三阶段执行学习操作：根据需要更新突触的连通值以及柱状区域的促进系数和抑制半径。

主要学习法则的实现在 20-26 行。对于胜利（因输入而最终变得活跃）的柱状区域，如果一个突触是活跃的，它的连通值就增加，反之就减少。连通值被限制在 0 到 1 之间。

28-26 行实现增强。有两个独立的促进机制以适当地帮助一个柱状区域学习如何连接。如果一个柱状区域不是经常被激活（通过 *activeDutyCycle* 衡量），那么它的促进系数就会增加（30-32 行）。与之相对应的，如果一个柱状区域的连通突触对任意输入的覆盖情况经常不是很好（通过 *overlapDutyCycle* 衡量），它的连通值将被促进增强（34-36 行）。注意：当学习被关闭时，*boost(c)* 被冻结。

最后，在第三阶段的末尾重新计算抑制半径（38 行）。

```
18. for c in activeColumns(t)
19.
20.     for s in potentialSynapses(c)
21.         if active(s) then
22.             s.permanence += permanenceInc
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.
```

支持的数据结构和程序

下面的变量和数据结构被用于伪代码中：

columns	所有柱状区域的列表。
input(t,j)	这一层级在 t 时刻的输入。当第 j 个输入为 1 时 input(t, j)为 1.
overlap(c)	柱状区域 c 对于特定输入的空间沉积池的覆盖值。
activeColumns(<i>t</i>)	因自底向上输入而活跃的柱状区域列表。
desiredLocalActivity	一个控制经过抑制后仍活跃的柱状区域数量的参数。
inhibitionRadius	被连接到的柱状区域的平均感受域大小。
neighbors(c)	在柱状区域 c 抑制半径内的所有柱状区域的列表。
minOverlap	会进入抑制过程的柱状区域的覆盖值的最小值。
boost(c)	在学习时被用于计算的柱状区域 c 的促进系数—用于增加非活跃柱状区域的覆盖值。
synapse	表示突触的数据结构—包含一个连通值和输入源索引。
connectedPerm	如果一个突触的连通值大于此值，那么就称它为连通的。
potentialSynapses(c)	一个潜在突触和它们的连通值的列表。
connectedSynapses(c)	连通值大于 connectedPerm 的 potentialSynapses(c)的子集。这些是当前自底向上输入连接到柱状区域 c 的突触。
permanenceInc	所有突触的连通值在学习时的增加量。
permanenceDec	所有突触的连通值在学习时的减少量
activeDutyCycle(c)	一个表示柱状区域 c 在抑制后的激活周期的变化的平均值（例如 1000 次的循环周期发生一次）

`overlapDutyCycle(c)` 一个表示柱状区域 `c` 对于其输入有效覆盖值（即大于 `minOverlap`）出现周期的变化的平均值（例如 1000 次的循环周期发生一次）。

`minDutyCycle(c)` 一个表示对细胞预期的最小使用率。如果一个细胞的使用率低于此值，那么它会被促进增强。这个值是附近最大的使用率的 1%。

下面的支持函数程序被用于之前的代码中：

`kthScore(cols, k)`

给定柱状区域列表 `cols`，返回第 `k` 高的覆盖值。

`updateActiveDutyCycle(c)`

计算柱状区域 `c` 在抑制后被激活的周期的移动平均数。

`updateOverlapDutyCycle(c)`

计算柱状区域 `c` 的覆盖值大于 `minOverlap` 的周期的移动平均数。

`averageReceptiveFieldSize()`

所有柱状区域连通感受域的平均半径。一个柱状区域的连通感受域只包含连通的突触（即连通值 \geq `connectedPerm`）。这个值用来决定柱状区域间的横向抑制的范围。

`maxDutyCycle(cols)`

给定柱状区域列表 `cols`，返回最高的柱状区域使用率。

`increasePermanences(c, s)`

通过一个标量 `s` 增加柱状区域 `c` 的所有突触的连通值。

`boostFunction(c)`

返回一个柱状区域的促进系数。促进系数是一个大于等于 1 的标量。如果 `activeDutyCycle(c)` 大于 `minDutyCycle(c)`，那么促进系数为 1。当柱状区域的 `activeDutyCycle` 开始小于 `minDutyCycle` 时，促进系数呈线性增长。

第四章：时间沉积的实现和伪代码

本章包含了一个时间沉积池功能的初步实现的详细伪代码。这段代码的输入是由空间沉积池计算出的 $activeColumns(t)$ 。代码计算的是每个细胞在当前时刻 t 的活跃和预测状态。每个细胞活跃和预测状态的逻辑或结果构成了时间沉积池对下一层级的输出。

伪代码按照发生顺序被分为三个不同的阶段：

阶段一：计算每个细胞的活跃状态， $activeState(t)$

阶段二：计算每个细胞的预测状态， $predictiveState(t)$

阶段三：更新突触

阶段三只是为了学习的需要。但是与空间沉积不同，阶段一和阶段二都包含了一些当处于学习状态的特有的学习操作。因此时间沉积比空间沉积要复杂得多，我们先给出一个只有识别功能的时间沉积池，然后再给出识别与学习组合的版本。对于实现细节、专业术语以及支持程序的描述在本章的末尾，伪代码之后。

时间沉积池伪代码：仅识别

阶段一

阶段一计算每个细胞的活跃状态。对于每个获胜的柱状区域我们决定哪些细胞应该变得活跃。如果自底向上的输入被任何细胞所预测到了（即它的 *predictiveState* 因前一时刻的横向序列片段而为 1），那么这些细胞就会变得活跃（4-9 行）。如果自底向上的输入是非预期的（即没有一个细胞的 *predictiveState* 为 1），那么柱状区域的每个细胞都会变得活跃（11-13 行）。

```
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
```

阶段二

阶段二计算所有细胞的预测状态。一个细胞的 *predictiveState* 会置为 1，当且仅当它的横向区域变得活跃，也就是说它的横向连接正因前馈输入而处于高使用率状态。

```
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
```

时间沉积池伪代码：识别与学习结合

阶段一

阶段一计算获胜的柱状区域中每个细胞的 *activeState*。对于这些柱状区域，代码进一步选择单细胞柱状区域的细胞作为学习细胞 (*learnState*)。下面是执行逻辑：如果自底向上的输入被任意细胞所预测（即它的 *predictiveState* 因树突区域的预测而为 1），那么这些细胞将被激活（23-27 行）。如果那个树突区域是由 *learnState* 为 1 的细胞激活的，那么这个细胞将被选为学习细胞（28-30 行）。如果自底向上的输入没有被预测到，那么所有的细胞将被激活（32-34 行）。另外，最匹配的细胞将被选为学习细胞（36-41 行）并且将一个新的树突区域添加到该细胞上。

```
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)
```

阶段二

第二阶段计算每个细胞的预测状态。一个细胞的预测状态输出 1 当且仅当它的横向连接（即树突区域）变得活跃，也就是说它的横向输入正在因前馈输入而活跃。在这种情况下，细胞排队等待一下改变：a）对当前活跃的横向连接增强（47-48 行），b）加强能预测当前激励的横向连接，即与之前时刻的活跃匹配（可能不是很匹配）的横向连接（50-53 行）。

```
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)
```

阶段三

第三和最后的阶段实际上执行学习操作。在这一阶段才会真正进行更新等待队列中的横向连接，当我们获得前馈输入并且细胞被标记为学习细胞（56-57 行）。不然，当细胞因为任何原因停止预测时我们会否定对横向连接的增强（58-60 行）。

```
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.
```

实现细节和术语

在这一部分我们讨论一些时间沉积池的实现细节和术语。每个细胞都是用两个数字作为索引：一个是柱状区域索引 c ，和一个细胞索引 i 。细胞维护着一个树突区域的列表，每个树突区域又包含一个突触的列表以及它们的连通值。细胞突触的改变被标记为临时的直到细胞因前馈输入而变得活跃。这些临时修改储存在 `segmentUpdateList` 中。每个树突区域还维护着一个布尔值标记，`sequenceSegment`，它指示了树突区域是否预测了下一刻的前馈输入。

潜在突触的实现与空间沉积池中的实现不同。在空间沉积池，完整的潜在突触列表用一个明确的列表表示。在时间沉积池，每个树突区域都有自己的（可能很大）潜在突触列表。实际上维护每个树突区域的长列表需要大量的计算和空间。因此在时间沉积池中，我们在学习过程中随机地添加活跃突触到树突区域（通过参数 `newSynapseCount`）。这种优化与维护所有潜在突触的列表有同样的效果，但是每个树突区域的列表在保证学习新临时模式的可能的同时还很小。

伪代码还使用了一个小自动机来记录细胞在不同时刻的状态。我们提出了三种不同的细胞状态。数组 `activeState` 和 `predictiveState` 记录了每个细胞在每个时刻的活跃和预测状态。数组 `learnState` 决定了哪个细胞的输出将在学习时使用。当一个输入是非预期的，某一特定柱状区域的所有细胞在同时刻变得活跃。这些细胞中只有一个（与输入最匹配的细胞）细胞的 `learnState` 会被置为 1。我们只对 `learnState` 为 1 的细胞添加突触（这避免了充分活跃的柱状区域有过多的树突区域表示）。

以下的数据结构被用于时间沉积池的伪代码中：

cell(c,i)	以 i 和 c 为索引的细胞列表。
cellsPerColumn	每个柱状区域的细胞数量。
activeColumns(t)	因自底向上输入而活跃的柱状区域索引列表（这是空间沉积池的输出）。
activeState(c, i, t)	每个细胞都有的一个布尔向量。它表示给定了当前前馈输入和之前的时间环境，柱状区域 c 的细胞 i 在 t 时刻的活跃状态。activeState(c, i, t)是柱状区域 c 的细胞 i 在 t 时刻的产出。如果为 1，那么细胞有当前的前馈输入和适当的时间环境。
predictiveState(c, i, t)	每个细胞都有的一个布尔向量。它表示了给定自底向上的其它柱状区域的激励和过去的时间环境，柱状区域 c 的细胞 i 在 t 时刻的预测。predictiveState(c, i, t)是柱状区域 c 的细胞 i 在 t 时刻的产出。如果为 1，那么细胞正在预测在当前时间环境下的前馈输入。
learnState(c, i, t)	一个指示柱状区域 c 的细胞 i 是否被选为学习细胞的布尔值。
activationThreshold	树突区域的活跃阈值。如果一个树突区域的活跃的连通突触数量高出活跃阈值，那么这个树突区域就被认为是活跃的
learningRadius	一个可以获得时间沉积池细胞横向连接的周围区域。
initialPerm	突触连通值的初始值。
connectedPerm	如果突触的连通值大于此值，那么就认为它是连通的。
minThreshold	树突区域学习的最小阈值。
newSynapseCount	在学习期间添加到树突区域的突触数量的最大值。
permanenceInc	所有突触在基于活跃的学习发生时的连通值增加量。
permanenceDec	所有突触在基于活跃的学习发生时的连通值减少量。

- `segmentUpdate` 更新给定树突区域的数据结构，拥有三部分信息：a) 树突区域索引（新树突区域为-1），b) 存在的活跃突触列表，c) 一个标记，指示当前树突区域是否被标记为 `sequence segment`（即预测了下一刻的前馈输入的树突区域），默认为 `false`。
- `segmentUpdateList` `segmentUpdate` 结构体的列表。`segmentUpdateList(c,i)`表示柱状区域 `c` 的细胞 `i` 的更新。

下面的支持函数程序被用于之前的代码中：

`segmentActive(s, t, state)`

当树突区域 `s` 在 `t` 时刻因给定的 `state` 而活跃的连通突触数量大于 `activationThreshold` 时，这个程序返回 `true`。参数 `state` 可以是 `activeState` 或 `learnState`。

`getActiveSegment(c, i, t, state)`

对于给定的柱状区域 `c` 的细胞 `i`，返回一个 `segmentActive(s,t, state)`为 `true` 的树突区域索引。如果有多个树突区域活跃，预测了下一刻输入的树突区域优先给出。不然就给出活跃度最高的树突区域。

`getBestMatchingSegment(c, i, t)`

对于给定的柱状区域 `c` 的细胞 `i` 在 `t` 时刻，搜索活跃突触最多的树突区域。这个程序在寻找最优匹配上表现得很积极。连通值小于 `connectedPerm` 的突触也会被考虑在内。活跃的突触可以小于 `activationThreshold`，但必须大于 `minThreshold`。程序返回树突区域索引。如果没有找到则返回-1。

`getBestMatchingCell(c)`

对于给定的柱状区域 `c`，返回含有最佳匹配树突区域（上面定义的）的细胞。如果没有细胞含有匹配树突区域，那么返回树突区域数量最少的细胞。

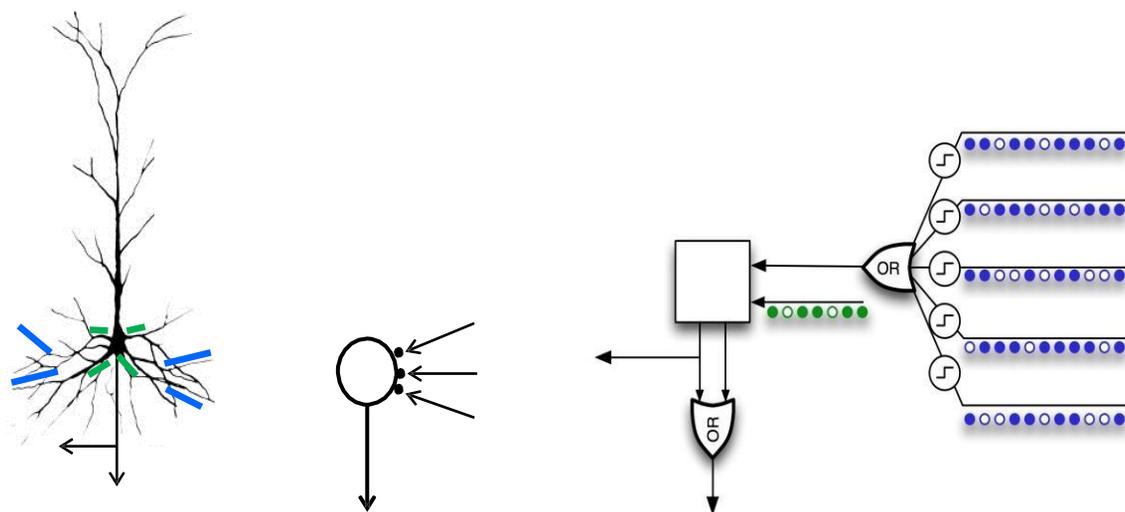
`getSegmentActiveSynapses(c, i, t, s, newSynapses= false)`

返回一个 `segmentUpdate` 数据结构，包含对树突区域 `s` 的修改计划列表。用 `activeSynapses` 作为活跃突触列表，当原细胞的 `activeState` 在 `t` 时刻为 1 时。（当 `s=-1` 时这个列表为空，因为树突区域不存在。）`newSynapses` 是可选参数，默认为 `false`。如果 `newSynapses` 为 `true`，那么 `newSynapseCount - count(activeSynapses)` 个突触会被添加到 `activeSynapses`。这些突触是随机从 `t` 时刻 `learnState` 为 1 的细胞集合中选出的。

`adaptSegments(segmentList, positiveReinforcement)`

这个函数循环访问 `segmentUpdate` 列表并对每个树突区域进行增强。对于每个 `segmentUpdate` 元素执行以下操作。如果 `positiveReinforcement` 为 `true`，那么在活跃突触列表中所有突触的连通值增加 `permanenceInc` 这么多。其它所有突触的连通值减少 `permanenceDec` 这么多。如果 `positiveReinforcement` 为 `false`，那么所有活跃突触列表中的突触的连通值都减少 `permanenceDec`。在这一步完成后，在 `segmentUpdate` 中还存在（连通值 \geq `connectedPerm`）的突触，其连通值加 `initialPerm`。

附录 A：生物神经元细胞与 HTM 细胞的对比



上面的图片中左边是生物神经元，中间是简单的人工神经元，右边是 HTM 神经元或者说是“细胞”。本附录的目的在于通过与真实的神经元和简单的人工神经元进行对比来更好地认识 HTM 的细胞是如何工作的。

真实的神经元是惊人的复杂并且种类繁多。我们只关注那些被用于我们模型中的普遍规律。尽管我们忽略了真实神经元的大量细节，但是 HTM 脑皮质学习算法中使用的细胞比起大多数神经网络中的人工神经元要真实得多。HTM 细胞中所包含的元素对于 HTM 区域的运算来说是必需的。

生物神经元

神经元是大脑中负责传输信息的细胞。上图中左侧的是一个典型的兴奋的神经元。神经元的视觉外观主要是树突分支在掌控。一个神经元所有的兴奋输入都会通过与树突成对的突触。近几年我们对神经元的认识取得了相当大的进步。最大的改变是认识到了神经元的树突并不只是将输入送到胞体的导管。我们现在知道了树突还是复杂的非线性处理单元。HTM 脑皮质学习算法正是利用这些非线性的特点。

神经元有几部分

细胞体

细胞体是神经元细胞中心的一小团东西。细胞的输出，轴突，起始于细胞体。细胞的输入通过与树突——对应的突触传输到胞体。

近体端树突

距胞体最近的树突分支被称为近体端树突。在上图中部分近体端树突用绿线表示。

近体端树突的多个活跃突触对于胞体有近似线性累加的作用。五个活跃突触会导致相对于一个活跃突触五倍的胞体去极化。与之相反，如果一个突触被一串连续的动作电位反复刺激，第二次，第三次和以后的动作电位对胞体的影响远小于第一次。

因此，我们可以说近体端树突的输入在胞体线性相加，连续到达单个突触的激励只会比单一激励的作用稍大一点。

新大脑皮层的前馈连接优先与近体端树突连接。实验表明至少作为每个区域的初级输入神经元层的第四层是这样的。

末端树突

距胞体距离较远的树突被称为末端树突。上图中的部分末端树突用蓝线表示。

末端树突比近体端树突更纤细。它们在树突树的分支末端与其它树突相连接并且不与胞体直接相连。这些不同赋予了末端树突独有的电学和化学特性。当末端树突的一个突触被激活，它对胞体的影响很小。发生于突触的极化反应等其到达了胞体已经被减弱。多年来这一直被视为一个谜。它使得占神经元大多数的末端树突看起来并没有什么实际作用。

我们知道末端树突部分与半独立的处理区域相似。如果在同一时刻有足够多的突触被激活并且在一个小范围内，它们就能产生一个会对胞体产生巨大影响的树突激励。例如，相互之间距离小于 40 微米的活跃突触会产生一个树突激励。因此，我们可以认为末端树突的行为像是一堆阈值符合探测器。

构成末端树突的突触主要来自区域中附近的其它细胞。

图中展示的巨大的向上拓展的树突被称为顶上树突。一种理论称这种结构使得神经元能够定位在同一区域内的几个末端树突，并且这个区域让它们能更容易地连接到传输的轴突上。根据这种解释，顶上树突就像扩展了的细胞。

突触

一个典型的神经元有几千个突触。它们中的大多数（90%左右）在末端树突上，其余的在近体端树突上。

多年以来人们都假设学习在于增强和削弱突触的作用或者说“权”。尽管这种作用已被人们观测到，但每个突触多少还存在一些随机性的东西。当它被激活，它不会非常可靠地释放出神经递质。因此大脑所使用的算法不可能依赖精确的或精准的个别突触的权值。

另外我们知道所有的突触形成和分解都很迅速。这就灵活地代表了一个强大的学习框架并且解释了学习知识的快速。只有当轴突和树突在一定范围之内才会形成突触，这便诞生了“潜在”突触这一概念。根据这些设想，学习主要是将潜在突触构建成有效突触。

神经元输出

神经元的输出像一个尖峰信号，或者是沿着轴突传送的“动作电位”。轴突离开胞体并且几乎总是分为两个分支。一条分支水平分布与其它附近的细胞建立许多连接。另一条分支连接到其它层的细胞或者大脑中的其它地方。在上面的神经元图像中轴突没有显示出来。我们用直线和两个箭头表示轴突。

尽管神经元的实际输出总是尖峰信号，但是对于如何解释它们有不同的观点。主要观点认为（尤其对于新大脑皮层来说）尖峰信号的频度至关重要。因此细胞的输出可以被视为一个标量。

一些细胞还表现出“爆发”的行为，一连串短促的尖峰信号与常规的尖峰信号模式不同。

以上关于神经元的描述旨在给出一个神经元的简介。它主要关注的是与 HTM 细胞相符合的特性并且省略了许多细节。不是所有我们描述的特性都是被广泛接受的。我们将其包含在内只是因为它们对于我们的模型是必需的。关于神经元的只是可以轻易填满好几本书，而关于神经元活动的研究持续至今。

简单的人工神经元

本附录起始处图片的中间展示了一种被广泛用于许多经典人工神经网络模型的类神经元元素。这些人工神经元有一个带权的突触集合。每个突触接收标量的激励，并且将其乘以突触的权值。所有突触的输出进行非线性加合以便产生人工神经元的输出。学习就发生在对突触权值或者是非线性函数的调整。

这种人工神经元及其变种被证明适用于许多应用问题，作为一个有价值的计算工具。但是它并不具有同生物神经元一般的复杂性和处理能力。如果我们想要理解并对大脑中真实神经元的工作机制建模，那么我们就需要一个更复杂的神经元模型。

HTM 细胞

在我们的例图中，右边的图像描绘了 HTM 脑皮质学习算法所使用的细胞。HTM 细胞具有真实神经元的许多重要能力但也进行了一些简化。

近体端树突

每个 HTM 细胞仅有一个近体端树突。细胞所有的前馈输入都要经过突触（用绿点表示）。突触的活跃线性加和后产生了细胞的前馈激励。

我们规定同一柱状区域的细胞有相同的前馈响应。在真实的神经元中这是由一种抑制细胞完成的。在 HTM 中我们简单地为一柱状区域的细胞指定一个共享的近体端树突。

为了避免有细胞在于邻近细胞的竞争中从未获胜的情况出现，相对于邻近细胞活跃较少的 HTM 细胞会增强它的前馈激励。因此细胞之间一直存在一个竞争。再次声明，在 HTM 网络中我们用柱状区域间的竞争来模拟细胞间的竞争。例图中没有对竞争进行描绘。

近体端树突有一个相关联的潜在突触集合，是整个区域输入的子集。在细胞学习期间，它增加或减少近体端树突所有潜在突触的连通值。只有那些高于阈值的潜在突触是有效的。

正如之前曾提到的，潜在突触这一概念源于生物学中轴突和树突足够近的时候可以构成突触。我们将这一概念扩展至更大的 HTM 细胞的潜在连接集合。生物神经元中的树突和轴突在学习时可以生长和回缩，因此潜在突触集合也要随着成长而变化。通过扩大 HTM 细胞的潜在突触集合，我们基本上得到了与生物轴突和树突相同的结果。潜在突触未在图中画出。

柱状区域间的竞争，潜在突触的学习，以及对为充分利用的柱状区域的刺激，这些组合在一起赋予了 HTM 神经元如大脑一般强大的可塑性。HTM 区域会自动调整每个柱状区域的表征（通过改变近体端树突的突触），当输入发生改变或者柱状区域的数量增加或减少。

末端树突

每个 HTM 细胞维护着一个末端树突列表。每个树突区域像一个阈值探测器。如果树突区域中活跃突触（前面图中的蓝点）的数量高于阈值，那么树突区域将变得活跃并且相连的细胞将进入预测状态。细胞的预测状态是树突区域激励的或运算结果。

树突区域通过为同一时刻活跃的细胞建立连接来记录区域的状态。树突区域记录细胞因前馈输入而变得活跃之前的状态。因此树突区域寻找着预测其所在细胞将变得活跃的状态。树突区域的阈值一般为 15。如果同一时刻树突区域中有 15 个有效突触活跃，那么树突将被激活。附近成百上千的细胞可能都处于活跃，但是连接到 15 个就足以识别较大的模式了。

每个末端树突也有一个相关联的潜在突触集合。潜在突触的集合是区域中所有细胞的子集。当树突区域学习的时候，它增加或减少其所有潜在突触的连通值。只有连通值高于阈值的潜在突触是有效的。

在一种实现中，我们固定每个细胞的树突区域数量。在另一种实现中，我们在训练中添加和删除树突区域。两种方法都有效。如果每个细胞有固定数量的树突区域，它可能在同一树突区域内有不同的几个突触集合。假设我们有 20 个有效突触在同一树突区域内并且阈值为 15。（一般我们将阈值设置地低于突触数量以增强抗噪性。）树突区域现在可以识别附近细胞的一

特定模式。如果我们将另外 20 个表示附近细胞完全不同的另一种模式的突触添加到同一树突区域会怎么样呢？这可能会导致错误，因为树突区域可能会因一个模式中的 8 个活跃突触和另一个模式中 7 个活跃突触（加起来 15 个达到阈值）而导致不正确的活跃。我们通过实验发现在发生错误之前至多在同一树突区域内存储 20 个不同的模式。因此一个有 12 个树突区域的 HTM 细胞可以参与到许多预测当中。

突触

HTM 细胞的突触有一位二进制权值。在 HTM 模型中并不排斥传统的标量突触权值，但是因为稀疏离散表征的应用使得我们没有用标量权值的必要。

然而，HTM 细胞的突触有一个标量称为“连通值”，它在学习过程中会不断调整。连通值为 0.0 表示一个潜在突触没有实际作用并且对于称为有效突触没有任何进步。低于阈值的连通值（如 0.2）表示一个突触刚刚建立连接但是连接很容易分解。连通值较高的（如 0.9），表示一个突触连通并且不容易被破坏。

HTM 细胞近体端和末端树突区域的有效突触数量是不固定的。当输入进入细胞时它会发生变化。例如，末端树突区域的有效突触数量依赖于数据的时间相关性。如果没有持续的时间连续的模式输入到区域，那么末端树突区域所有突触的连通值都会很低并且只有少量的有效突触。如果输入流中存在大量时间上相关的模式，那么我们会发现有许多有效突触并且连通值都较高。

细胞输出

一个 HTM 细胞有两种不同的二进制输出：1）因前馈输入而产生的兴奋（通过近体端树突），和 2）因横向连接而产生的兴奋（通过末端树突区域）。前者称为“活跃状态（active state）”，后者称为“预测状态（predictive state）”。

在先前的图中，两种输出用两条离开正方体胞体的直线表示。左边的线是前馈活跃状态，右边的是预测状态。只有前馈活跃状态会连接到同一区域内的其它细胞，以确保预测总是基于当前输入（和环境）。我们不希望基于预测而预测。如果我们这样做，那么在几次循环后几乎区域内的所有细胞都会进入预测状态。

区域的输出是一个表示所有细胞活跃状态的向量。这个向量是层级结构中下一个区域的输入，如果其存在。这个输出时活跃状态和预测状态逻辑或的结果。通过将活跃状态和预测状态组合在一起，我们区域的输出会比输入更稳定（变化慢）。这种稳定性是区域识别的重要特性。

推荐读物

经常有人让我们推荐些学习神经系统科学的阅读材料。神经系统科学的范围太大了以至于一个大概的了解都需要看很多不同来源的资料。新发现总是发表在学术期刊上，这些比较难读懂并且也不容易得到，如果你与大学的研究机构没有关系。

这里给出两本容易理解的书，喜欢钻研的读者可以阅读本附录中相关话题的内容。

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

这本书是关于树突的一个较好的初级读物。第 16 章讨论了 HTM 脑皮质学习算法使用的树突区域的非线性特性。它是由 Bartlett Mel 所著，他在其领域颇有建树。

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

这本书很好地介绍了新大脑皮层的相关知识。有几章讨论了细胞类型及其连接。你可以借此对脑皮质神经元和连接有个更好的认识，尽管它有点旧以至于不能包含最近的树突特性的相关知识。

附录 B：新大脑皮层中的层与 HTM 中的层的对比

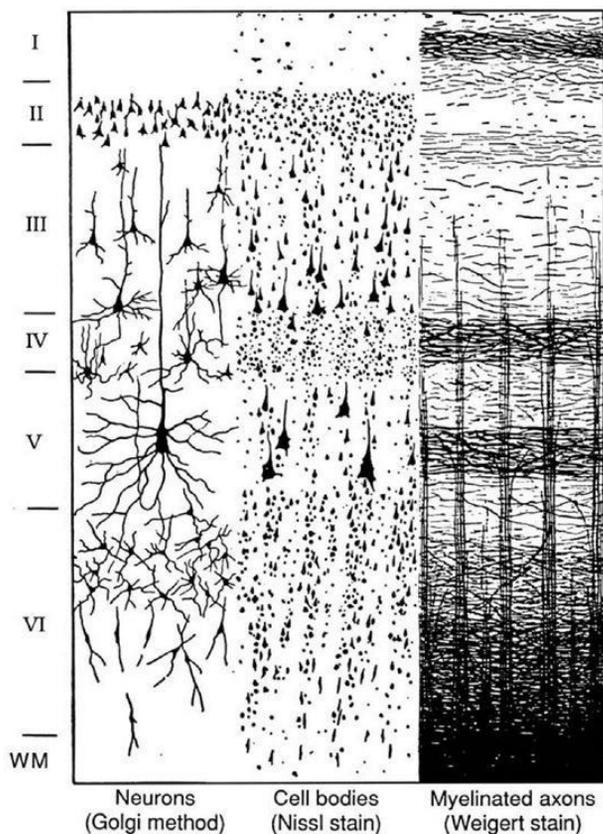
本附录主要介绍 HTM 区域与生物学上新大脑皮层区域之间的关系。

值得注意的是，本附录包含了 HTM 脑皮质学习算法的柱状区域和细胞与新大脑皮层层状及柱状结构有怎样的联系。许多人对新大脑皮层中的“层”这一概念和其与 HTM 的层之间的关系感到困惑。希望本附录能解决这种疑惑并且能够使你对 HTM 脑皮质学习算法的生物学基础有更深刻的理解。

新大脑皮层的神经回路

人类的新大脑皮层是一层很薄的神经元组织，面积约 1000 平方厘米，厚度为 2 毫米。为了将其表示地更形象，你可以想象一下一张餐巾，其面积和厚度接近新大脑皮层的大小。新大脑皮层被划分为几十个功能区域，一些与视觉有关，另一些与听觉有关，还有与语言有关的等等。在显微镜下观察会发现，不同区域的物理特性有着惊人的相似性。

在新大脑皮层中有几个适用于所有区域的组织原则。



层

新大脑皮层通常被认为有 6 层。其中 5 个层包含细胞，剩余的一层主要是连接。随着染色法的出现，层在 100 多年前就被世人发现了。上图（来自 Cajal，西班牙著名的神经解剖学专家）展示了用三种不同的染色法对新大脑皮层切片处理后的效果。纵轴贯穿约 2mm 厚的新大脑皮层。左边的罗马数字表示 6 个层。第一层在顶端，是无细胞层。底端的“WM”表示脑白质的起始处，细胞的轴突从这里延伸到新大脑皮层和大脑的其它部分。

图中最右边只显示有髓鞘的轴突。（髓鞘是一种裹覆部分轴突的脂鞘。）在这部分图中你可以见到两种新大脑皮层的主要组织形式，层和柱状区域。大多数轴突在离开胞体后迅速一分为二。一个分支会向水平方向延伸，另一个会向垂直方向延伸。水平分支会与同层或邻层的细胞构成大量连接，因此层在染色法图中就是这样子。你需要牢记这个图只是新大脑皮层的一个切片。有大量的轴突垂直于图像所在的平面，因此轴突要比图中所示的长得多。根据估算，新大脑皮层中每立方毫米的轴突和树突长度在 2 至 4 千米之间。

图的中间部分是神经元胞体的染色图，但是没有显示任何树突和轴突。你可以发现神经元的大小和密度随着层的变化而变化。在这部分图像中只表现出一点柱状区域的迹象。你可能注意到第一层当中有一些神经元。但是这些神经元的数量极其少以至于第一层被认为是没有神经元的层。神经科学专家估测，每立方毫米新大脑皮层中有大约 100000 个神经元。

图的左边仅是几个神经元的胞体，轴突和树突的染色图。你会发现树突的“枝干”大小随着层的变化而发生显著地改变。你也可以看到一些从胞体向上延伸的“顶上树突”与其它层形成连接。不同层的顶上树突有不同的形态和延伸的终点。

简而言之，新大脑皮层的层状和柱状组织形式在染色法的帮助下变得十分明显并且能够在显微镜下观察。

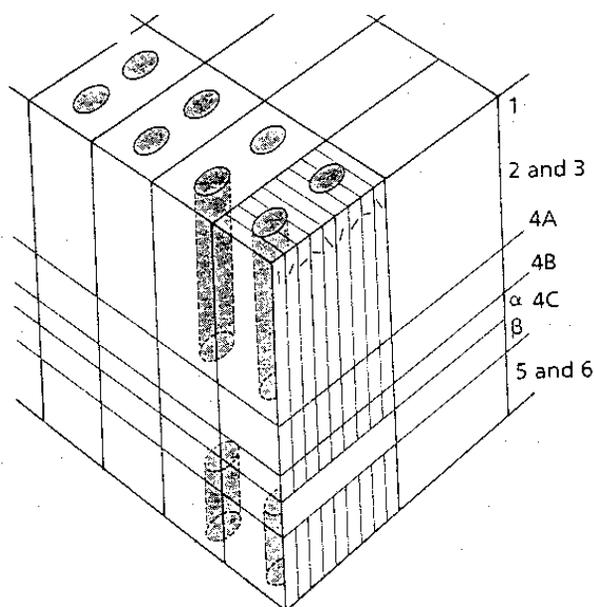
不同区域中层的变种

对于不同区域中层的厚度存在不同说法，并且围绕层的数目也有争议。这些不同的观点取决于用哪种动物进行实验，观察的是哪个区域，以及谁进行的实验。例如上图中第二层和第三层看起来很容易区分开，但通常并不是如此。一些科学家称他们无法在他们所研究的区域中区分这两层，所以第二层和第三层通常被一起称为“第 2/3 层”（将层合并）。另一些科学家走上了与之相反的方向，定义了如 3A 和 3B 这类的子层（将层分割）。

第四层在靠近感觉器官的新大脑皮层区域中被较好的定义。对于一些动物（如人类和猿类），第一视觉区域（V1）中的第四层是很容易区分出的。但对于另一些动物就不容易区分。在层级结构中远离感觉器官的区域基本上没有第四层。

柱状区域

新大脑皮层第二重要的组织形式就是柱状区域。一些柱状组织在染色法图像中能观察到，但是大多数柱状区域的证据都是基于细胞如何响应不同的输入。当科学家用探针去测试什么使得神经元活跃的时候，他们发现神经元是纵向对齐的，贯穿了不同的层，并对大致相同的输入产生响应。



本图中展现了 V1 区细胞的响应特性，它是处理由视网膜传入信息的第一个区域。

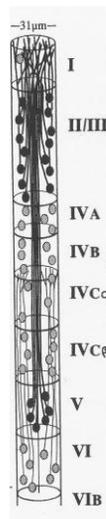
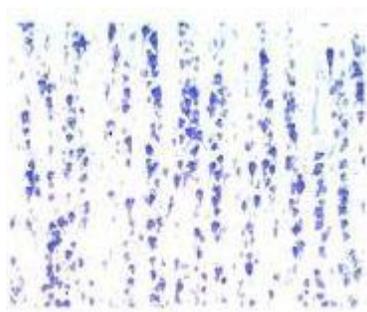
首当其冲的发现是 V1 区的多数细胞对视网膜特定区域的不同朝向的直线或边产生响应。同一柱状区域中垂直对齐的细胞对相同朝向的边产生响应。如果你看得够仔细，你会发现图中上面的部分排列着一系列不同朝向的小直线。这些直线反映了所在柱状区域细胞所响应直线的朝向。垂直对齐的细胞（在同一垂直细条中）对相同朝向的直线产生响应。

V1 区域还表现出柱状区域的另一些特性，其中两个在图中有所显示。存在所谓的“眼优势柱”（支持立体深度感知）对左右眼中的相似组合产生响应。对颜色敏感的细胞被称为“斑点”视觉统一区域是图中较大的块状物。每个眼优势柱包含一系列表示不同朝向的柱状区域。“斑点”是灰色的椭圆柱。

新大脑皮层的一个通则是不同方向的响应内容会相互重叠，例如朝向和眼优势。当你水平地观察皮质表面时，随着细胞的改变响应内容的集合被展现出来。然而垂直对齐的细胞也共享着相同的响应内容集合（即上图中较薄的纵断面）。这种纵断面存在于听觉、视觉及触觉区域中。在神经科学家中对新大脑皮层中是否到处都有这种纵断面留有争议，但是即使不是全部，对于大多数区域来说也有这种纵断面。

迷你柱状区域

迷你柱状区域是新大脑皮层中最小的柱状结构。迷你柱状区域直径约 30 微米并且包含 80-100 个神经元贯穿 5 个有细胞的层。整个新大脑皮层就是由迷你柱状区域组成的。你可以把它们想象成一条条细的意大利面堆放在一起。迷你柱状区域之间有几个细胞大的细小空隙，这使得它们有时能从染色图中观察到。



左边是新大脑皮层局部切片中的神经元胞体染色图。迷你柱状区域的垂直结构在这张图中得到证明。右边的是迷你柱状区域的概念图（来自 Peters 和 Yilmaz）。实际中的要比这个还细小。注意柱状区域中的每层有多个神经元。迷你柱状区域的所有神经元对相似的输入产生响应。例如在之前所示的 V1 区局部图中，迷你柱状区域包含对有特定眼优势倾向的特定朝向的直线产生响应的细胞。相邻迷你柱状区域的细胞可能对朝向和眼优势稍不同的直线产生响应。

抑制神经元扮演着定义迷你柱状区域的关键角色。它们在图中不可见，但是抑制神经元将迷你柱状区域用笔直的轴突连接起来以便对其进行物理上的分离。抑制神经元还被认为有助于约束同一迷你柱状区域的细胞对相似输入产生响应。

迷你柱状区域是 HTM 脑皮质学习算法中使用柱状区域的原型。

柱状区域响应的异常

对于柱状区域的响应有一个异常情况与 HTM 脑皮质学习算法有关。通常科学家通过对实验动物输入一个简单刺激来确定细胞对什么产生响应。例如，他们可能会向实验对象展示视觉空间中很小的一条直线来确定 V1 区域中细胞的响应内容。当用相同的输入时，研究者发现那个细胞总算是相同的输入产生响应。然而如果简单的输入被嵌入到一段自然场景的视频中，细胞的响应就变得不稳定了。对单独的直线准确产生响应的细胞面对嵌入到复杂运动图像中的直线不会总产生响应。

在 HTM 脑皮质学习算法中，同一柱状区域中的所有细胞享有相同的前馈响应内容，但是在一个已经习得的时间序列中，对于每个 HTM 柱状区域只有一个细胞变得活跃。这种机制是为了表示可变源序列并且与之前所述的神经元特性类似。一个简单的没有上下文环境的输入会导致整个柱状区域的细胞活跃。而在已习得序列中的相同输入只会造成一个细胞活跃。

我们并不是在暗示一个迷你柱状区域同一时间只有一个神经元会活跃。对于 HTM 脑皮质学习算法，这些发生在一个柱状区域内，同层中所有的细胞会被非预期的输入激活而对于预期中的输入只有一部分神经元被激活。

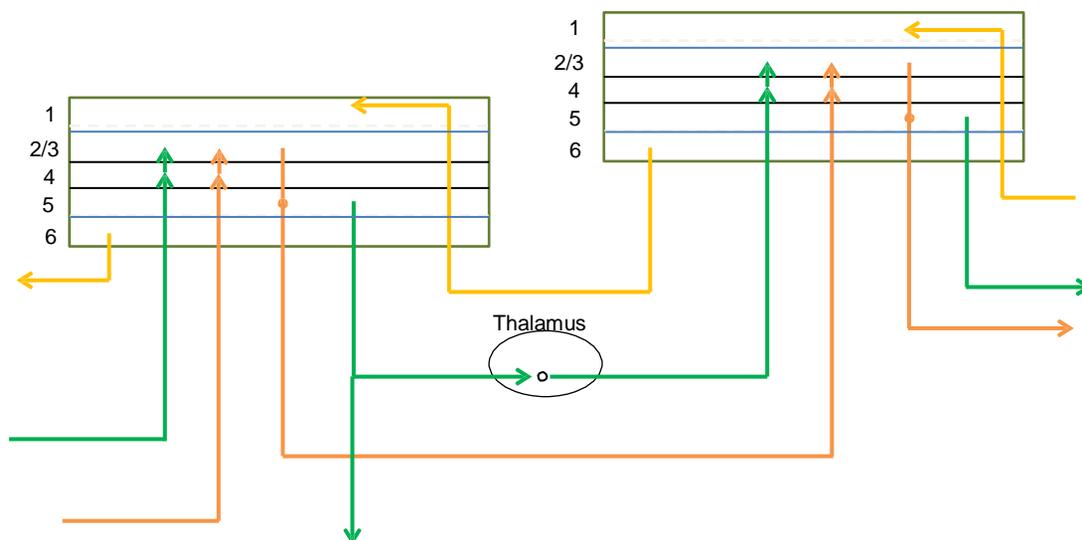
为什么会有层和柱状区域？

没人确切地知道新大脑皮层中为什么会有层和柱状区域。然而 HTM 理论提出了一种假设。HTM 脑皮质学习算法展现了柱状区域中的一层细胞所具有的高容量记忆空间，用来存储可变源的渐变状态。简而言之，一层细胞可以学习很多的序列。柱状区域细胞共享相同的前馈响应是学习可变源渐变的关键机制。

这个假设解释了为什么柱状区域是必需的，但是那 5 层呢？如果单个的脑皮质层可以学习序列和做出预测，为什么我们会看到新大脑皮层中有 5 层呢？

我们推断新大脑皮层中观察到的不同层用相同的基本机制学习序列但是每一层习得的序列的用途不同。这其中还有许多我们没有搞清楚的东西，但是我们可以把大体上的想法阐述一下。在

我们说明之前，介绍一下每层神经元是如何连接的会比较有帮助。



上图中描绘了两个新大脑皮层区域之间的主要连接。这些连接在新大脑皮层中相互映射的区域之间随处可见。左边的框表示一个皮层区域，它相对于右边的区域（框）处于层级结构中较低的位置，所以前馈信息从左边传到右边。向下的箭头映射到大脑中的其它区域。反馈信息从右往左传递。每个区域被划分成许多层。第二层和第三层以“第 2/3 层”的形式给出。

彩色的线条表示不同层神经元的输出。它们是起始于所在层神经元的轴突束。你要记得轴突会立即分成两条。一条分支在区域内水平延伸，主要在同一层。因此同一层内的细胞是高度内联的。神经元和水平连接没有在图中显示。

有两条前馈通路，橙色的线表示直接通路，绿色的线表示间接通路。第四层是初级前馈输入层并且接收来自两条前馈通路的输入。第四层映射到第三层。

第三层同样也是直接通路的起点。所以直接通路被限制在第三层和第四层。

有些前馈连接跳过第四层直接指向第三层。正如之前所提到的，第四层在离感官输入较远的区域消失了。在这种情况下，直接前馈通路只是从第三层到下一区域的第三层。

第二条前馈通路（绿线）起始于第五层。第三层在传入到下一区域时与第五层建立一些链接（图中的点）。第五层的轴突在离开新大脑皮层后再次分裂。一条分支映射到大脑皮层下与运动产生有关的区域。这种轴突被认为是运动的命令（图中笔直向下的箭头）。另一条分支映射到大脑中被称为丘脑的部分，它与门的作用相同。丘脑可能会向下一区域传递信息或者拦截信息。

最后是主要的反馈通路，用黄色表示，起始于第六层映射到第一层。第 2/3 层和第五层的细胞通过其顶上树突连接到第一层（图中未画出）。第六层接收第五层的输入。

以上是对层与层之间连接的已知知识的一个简要概述。但是对于理解我们关于为什么所有层都能学习序列还有多个层存在的假设来说已经足够了。

不同层功能的假设

我们认为第三、四、五层均为前馈层并且都学习序列。第四层学习单一源序列。第三层学习可变源序列。第五层学习时序可变源序列。让我们逐个详细讨论。

第四层

使用 HTM 脑皮质学习算法可以很容易地学习单一源序列。如果我们不强制要求同一柱状区域的细胞互相抑制，也就是说，同一柱状区域的细胞不区分之前输入的环境，那么就会发生对单一源序列的学习。在新大脑皮层中，这可能是通过移除同一柱状区域细胞之间的抑制作用来完成的。在我们的 HTM 脑皮质学习算法的计算机模型中，我们为每个柱状区域设置一个细胞以达到相同的效果。

单一源序列是对于空间变换的输入构建恒定表征所必需的。以视觉为例，x-y 转换，按比例缩放和旋转都是空间变换。当用运动的物体去训练单一源记忆的 HTM 区域时，它会认为不同的空间模式是等价的。结果 HTM 细胞的行为会像新大脑皮层中的“复杂细胞”。HTM 细胞会在经过一系列空间变换后仍保持活跃（处于预测状态）。

我们已经通过实验证明这种工作机制正如预料的那样，并且在每一层都实现了空间的稳定性。这些实验的详细细节不在本附录讨论范围之内。

第四层中对单一源序列的学习与在第四层中发现复杂细胞有密切关系，并且解释了为什么在新大脑皮层中较高的区域第四层消失的问题。随着层级结构的上升到一定程度，当表征已经趋于稳定，进一步学习空间恒定表征变得不再可能。

第三层

第三层最接近我们在第二章中对 HTM 脑皮质学习算法的描述。它学习可变源序列并且构建比输入更稳定的预测。第三层一般映射到层级结构中的下一级区域（较高的）因此导致了层级结构内在的时间稳定性。可变源序列记忆导致了在第三层中首次观察到了“方向校准复杂细胞”（directionally-tuned complex cell）。方向校准复杂细胞区分时间环境，例如朝右移动的直线和朝左移动的直线是不同的。

第五层

第五层是最终的前馈层。我们认为第五层与第三层大体相似，除了有三点不同。第一点不同是第五层增加了时序的概念。第三层预测了“什么”会发生但没有告诉你“什么时候”会发生。然而，许多任务都是对时序有要求的，例如对于识别语音来说，声音之间相对的时序是重要的。运动行为也是个例子，肌肉激励之间协调的时序也是关键。我们认为第五层的神经元只有在预定时间之后才会预测下一状态。有几个生物学上的细节可以支持这一假设。一个是第五层是新大脑皮层的运动输出层。另一个是第五层接收第一层的输入，起源于丘脑的一部分（图中未画出）。我们认为时间信息就是这样通过丘脑输入到第一层并编码、分布许多细胞当中的。

第三层和第五层之间的第二点不同是我们希望第三层所做出的预测尽可能地远，以便获得时间稳定性。第二章描述的 HTM 脑皮质学习算法正是如此。与之相反，我们只希望第五层预测下一个状态（在特定的时刻）。我们没有针对这一不同点建模，但是当渐变总是以时间相关的形式被储存，那么它就会自然而然地发生。

第三层和第五层的第三点不同可以在图中看到。第五层的输出总是映射到皮质下的运动中枢，并且前馈通路是由丘脑作为控制门。第五层的输出有时会传到下一区域，有时会被拦截。我们（和其他人）认为这种阀门控制与掩蔽注意有关（掩蔽注意是指在关注输入时不关注运动行为）。

总而言之，第五层包含了特定的时序，关注点和运动行为。关于它们是如何一起运作的还有许多未解之处。我们的目标是构建一个 HTM 脑皮质学习算法的变种，能够较容易地融进特定时序并通过脑皮质中单独一层的验证。

第二层和第六层

第六层是反馈回较低区域的轴突的起点。关于第二层知道的就更少了。正如之前所述，关于第二层是否独立于第三层存在一直存有争议。我们现在不会深入讨论这一问题，我们只会指出第二层和第六层同其它层相似，具有大量的横向连接模式和柱状响应特性，所以我们也打算把它们加入到 HTM 脑皮质学习算法的新变种当中去。

HTM 区域相当于新大脑皮层中的什么？

我们已经实现了两种版本的 HTM 脑皮质学习算法，一种是针对可变量记忆的多细胞柱状区域版本，另一种是针对单一源记忆的单细胞柱状区域版本。我们认为这两个版本相当于新大脑皮层中的第三层和第四层。我们没有试图将这两个变种融合到一个 HTM 区域中。

尽管 HTM 脑皮质学习算法（多细胞柱状区域的）最接近新大脑皮层中的第三层，但我们的模型有大脑所不具有的灵活性。因此我们可以创建混合的细胞层，它不是对应的新大脑皮层中的某一层。例如，在我们的模型中指令来源于树突区域中的突触。利用这些信息我们可以从未来所有可能发生的情况当中提取出接下来会发生的情况。我们可以把特定时序通过这样的方式加入进去。因此创建一个拥有第三层和第五层组合功能的单一 HTM 区域层应该是可行的。

综述

HTM 脑皮质学习算法实现了我们所认为的新大脑皮层中神经组织的基本架构。它展现了一层横向连接的神经元是如何学习稀疏离散表征序列的。不同的 HTM 脑皮质学习算法版本被用于描述新大脑皮层中不同的层，但是它们的设计目的是不同的。

我们认为一个新大脑皮层的前馈输入，无论是第三层还是第四层，都主要映射到近体端树突，在抑制细胞的协助下构建输入的稀疏离散表征。我们认为第二至六层享有共同的稀疏离散表征。这是通过限制贯穿数层的同一柱状区域的细胞对相同前馈输入相应而实现的。

我们认为第四层在 HTM 脑皮质学习算法中被用于学习单一源临时渐变，它能使表征对于

空间变换趋于稳定。第三层在 HTM 脑皮质学习算法中被用于学习可变源临时渐变并且构建传往皮层中更高级区域的表征。第五层学习时序可变源渐变。我们对第二层和第六层没有特别设定。但是它们与其它层具有相同的典型横向连通性，因此它们也在学习某种形式的序列记忆。

术语表

注意：这里术语的定义仅限于本文档内使用，它们可能有一般广泛使用的意思。加粗的术语（释义中的）指的是本术语表中定义的术语。

活跃状态	当 细胞 被 前馈输入 激活后的状态
自底向上	前馈 的同义词
细胞	HTM 中 神经元 的等价物 细胞 组成了 HTM 区域中的柱状区域。
同步活跃	多个 细胞 在同一时间被激活
柱状区域	由一个或多个 细胞 组成的 HTM 区域 的功能单元 同一柱状区域的 细胞 表示相同的前馈输入，但处于不同环境下。
树突区域	连接到 细胞 和 柱状区域 的 突触 的集合单元 HTM 有两种不同的树突区域。一种是横向连接到其它细胞。当树突区域中活跃突触的数量超过了阈值，相关联的细胞会进入预测状态。另一种是前馈连接到其它柱状区域。活跃突触数量的合计用来产生一个柱状区域的前馈激励。
期望密度	期望中的因 区域前馈输入 而活跃的 柱状区域 的百分比。 这个百分比仅用于一个基于前馈输入扇出而变化的半径内。它是“期望的”是因为百分比是根据特定输入而变化的。

前馈	从输入的方向传递而来，或者从 层级结构 中的 低层级 到 高层级 （有时称为 自底向上 ）
反馈	朝着输入的方向传递，或者从 层级结构 中的 高层级 到 低层级 （有时称为 自顶向下 ）
单一源预测	仅基于当前输入且与先前输入无关的预测—相对于 可变源预测
层级实时记忆（HTM）	一种模仿新大脑皮层结构和算法功能的技术
层级结构	一种由元素连接而成的网络，元素间的连接被唯一地识别成 前馈 或者 反馈
HTM 脑皮质学习算法	由组成一个 HTM 区域 的 空间沉积 、 时间沉积 、 学习 和 遗忘 所构成的功能体系，也被称为 HTM 学习算法
HTM 网络	由 HTM 区域 所构成的 层级结构
HTM 区域	HTM 中 记忆 和 预测 的主要单元 一个 HTM 区域由一层高度连通并且柱状排列的细胞所构成。现在的 HTM 区域只有一层细胞，然而在新大脑皮层（以及最终的 HTM 网络）中，一个区域会有多层细胞。当在讨论它在层级结构中所处于的位置时，一个区域指的是一个层级。
识别	将空间和时间的输入模式识别成之前所学过的模式
抑制半径	定义了一个 柱状区域 周围对活跃抑制的区域
横向连接	在同一 区域内 细胞 之间的连接
层级	在对 层级结构 讨论中的 HTM 区域

神经元	<p>大脑中处理信息的细胞</p> <p>在本文档中，我们用“神经元”特指生物学上的细胞，用“细胞”来指 HTM 的计算单元。</p>
连通值	<p>一个指示潜在突触连接状态的标量</p> <p>当连通值低于阈值时说明突触没有形成。当连通值高于阈值时说明突触是有效的。HTM 区域的学习就是通过修改潜在突触的连通值来完成的。</p>
潜在突触	<p>存在与特定树突区域形成突触可能的细胞子集</p> <p>潜在突触中只有一部分会在任意时刻成为有效突触，这要看它们的连通值。</p>
预测	<p>在近期可能因前馈输入而变得活跃的细胞（处于预测状态）</p> <p>HTM 区域经常会同时预测许多未来的可能输入。</p>
感受域	<p>输入所连接到的细胞或柱状区域的集合</p> <p>如果 HTM 区域的输入是二维二进制位的组织形式，那么感受域就可以表示成一个输入空间上的范围半径。</p>
感受器	HTM 网络的输入源
稀疏离散表征	一种由许多二进制位组成的表征，只有很小的一部分是“1”并且单个位表示不了具体意义

空间沉积	<p>构建输入的稀疏离散表征的过程</p> <p>空间沉积的特性之一是把重叠的输入映射到同一稀疏离散表征。</p>
子抽样	通过只比较巨大模式中活跃位的一小子集来识别巨大的离散模式
突触	在学习过程中构建的 细胞 之间的连接
时间沉积	输入序列构建表征的过程，产生的表征比输入要更稳定。
自顶向下	反馈 的同义词
可变源预测	<p>一种基于大量变化的前提环境的预测——与单一源预测相对</p> <p>它之所以被称为“可变”是因为维持前提环境的记忆空间是按需分配的。因此实现可变源预测的记忆系统能够及时地调用之前的环境，并且不需要指数级的记忆空间。</p>