



MEMÓRIA TEMPORAL HIERÁRQUICA

incluindo

Algoritmos de Aprendizagem Cortical HTM

VERSÃO 0.2, 10 DE DEZEMBRO DE 2010

© Numenta, Inc. 2010

O uso de software Numenta e propriedade intelectual, incluindo as idéias contidas neste documento, são livres para fins de pesquisa não-comercial. Para mais detalhes, consulte <http://www.numenta.com/about-numenta/licensing.php>.

Tradução de David Ragazzi Rocha

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.

Licença de Tradução Numenta

Copyright (c) 2010, 2011 Numenta, Inc.

Todos os direitos reservados.

Os textos, algoritmos, códigos de exemplo, pseudo-códigos e outros trabalhos aqui incluídos são baseados ou traduzidos de certas obras relacionadas com a tecnologia Memória Temporal Hierárquica ("HTM") publicada pela Numenta Inc. A Numenta detém os direitos autorais sobre a obra original e direitos de patente relacionados com a HTM e os algoritmos traduzidos aqui. A Numenta concordou em não fazer valer seus direitos de patente contra o desenvolvimento ou o uso de sistemas independentes HTM, desde que tal desenvolvimento ou o uso seja apenas para fins de pesquisa, e não para qualquer uso comercial ou de produção. Qualquer utilização comercial ou de produção da tecnologia HTM que infrinja as patentes Numenta irá requerer uma licença comercial da Numenta.

Com base no exposto, a Numenta lhe concede uma licença para o uso destes algoritmos e obras para fins de pesquisa, e não para qualquer uso comercial ou de produção. Para os propósitos desta licença, o "uso comercial ou de produção" inclui treinar uma rede HTM com a intenção de posteriormente implantar a rede treinada ou o aplicativo para fins comerciais ou de produção, e usar ou permitir que outros utilizem a saída da tecnologia HTM para fins comerciais ou de produção. Qualquer distribuição, publicação ou cópia desta obra deve incluir o texto completo desta Licença de Tradução em inglês e na língua-alvo.

LICENÇAS NÃO EXPRESSAS OU IMPLÍCITAS PARA QUAISQUER DIREITOS DE PATENTE SÃO CONCEDIDOS POR MEIA DESTA LICENÇA. A NUMENTA REPUDIA ESPECIFICAMENTE QUALQUER OBRIGAÇÃO OU RESPONSABILIDADE PELA QUALIDADE OU EXATIDÃO DE QUAISQUER TRADUÇÕES LICENCIADAS.

Leia Isto Primeiro!

Esta é uma versão preliminar deste documento. Há várias coisas faltando que você deve estar ciente.

O que ESTÁ neste documento:

Este documento descreve em detalhes novos algoritmos de aprendizagem e de predição desenvolvidos por Numenta em 2010. Os novos algoritmos são descritos em detalhe suficiente para que um programador possa entender e aplicar, se desejar. Ele começa com um capítulo introdutório. Se você tem acompanhado a Numenta e leu alguns dos últimos artigos, o material no capítulo introdutório será familiar. O outro material é novo.

O que NÃO está neste documento:

Há vários tópicos relacionados com a implementação destes novos algoritmos que não tornam este projeto inicial.

- Embora a maioria dos aspectos dos algoritmos tenha sido implementada e testada em software, nenhum dos resultados dos testes estão incluídos.
- Não há nenhuma descrição de como os algoritmos podem ser aplicados a problemas práticos. Está faltando uma descrição de como você poderia converter dados de um sensor ou banco de dados em uma representação distribuída adequada para os algoritmos.
- Os algoritmos são capazes de aprender on-line. Alguns detalhes necessários para a total implementação de aprendizagem on-line, em alguns casos mais raros não são descritos.
- Outras adições planejadas inclui uma discussão sobre as propriedades das representações distribuídas esparsas, uma descrição das aplicações e exemplos, e citações dos apêndices.

Estamos deixando este documento disponível em sua forma atual, porque nós pensamos que os algoritmos serão de interesse para os outros. Os componentes em falta do documento não devem impedir a compreensão e experimentação com algoritmos por pesquisadores motivados. Vamos rever este documento periodicamente para refletir o nosso progresso.

Sumário

Prefácio	6
Capítulo 1: Visão Geral da HTM	9
Capítulo 2: Algoritmos de Aprendizagem Cortical HTM	20
Capítulo 3: Pseudocódigo e Implementação do Agrupamento Espacial	34
Capítulo 4: Pseudocódigo e Implementação do Agrupamento Temporal	39
Apêndice A: Uma Comparação entre Neurônios Biológicos e Células HTM	45
Apêndice B: Uma Comparação das Camadas no Neocórtex e em uma Região HTM	51
Glossário	61

Prefácio

Existem muitas coisas que os seres humanos acham fáceis de fazer que os computadores são atualmente incapazes de fazer. Tarefas como reconhecimento de padrões visuais, compreensão da linguagem falada, reconhecimento e manipulação de objetos pelo tato, e navegar em um mundo complexo são fáceis para os seres humanos. No entanto, apesar de décadas de pesquisas, temos poucos algoritmos viáveis para alcançar o desempenho humano em um computador.

Em humanos, essas habilidades são em grande parte desenvolvidas pelo neocórtex. A Memória Temporal Hierárquica (HTM) é uma tecnologia modelada sobre o modo como o neocórtex desempenha essas funções. A HTM oferece a promessa de construção de máquinas que se aproximam ou excedem o nível de desempenho humano para diversas tarefas cognitivas.

Este documento descreve a tecnologia HTM. O capítulo 1 fornece uma visão ampla da HTM, destacando a importância da organização hierárquica, representações distribuídas esparsas, e transições baseada no tempo de aprendizagem. O capítulo 2 descreve os algoritmos de aprendizagem cortical HTM em detalhes. Os capítulos 3 e 4 fornecem pseudocódigo para os algoritmos de aprendizagem HTM dividido em duas partes chamadas de agrupador espacial e agrupador temporal. Depois de ler os capítulos 2 a 4, os engenheiros de software experientes devem ser capazes de reproduzir e experimentar com os algoritmos. Felizmente, alguns leitores irão mais além e alargar o nosso trabalho.

Público-alvo

Este documento é destinado a um público tecnicamente competente. Embora nós não assumamos que você tenha conhecimento prévio de neurociência, assumimos que você possa entender conceitos de matemática e ciência da computação. Nós escrevemos este documento de tal forma que o mesmo pode ser usado como leitura administrada em uma aula. O nosso leitor principal imaginado é um estudante de ciência da computação ou ciência cognitiva, ou um desenvolvedor de software que está interessado em construir sistemas cognitivos artificiais que funcionam nos mesmos princípios que o cérebro humano.

Os leitores não-técnicos podem ainda se beneficiar de certas partes do documento, especialmente o *Capítulo 1: Visão Geral da HTM*.

Liberação de software

Temos intenção de lançar o software baseado nos algoritmos descritos neste documento, em meados de 2011.

Relação com os documentos anteriores

Partes da teoria HTM são descritas no livro de 2004 *Sobre a Inteligência*, em documentos oficiais publicados pela Numenta, e em artigos de revistas escritos por funcionários Numenta. Nós não assumimos que você tenha lido estes materiais antes, muitos dos quais foram incorporados e atualizados neste volume. Note que os algoritmos de aprendizagem HTM descritos nos capítulos 2-4 não foram publicados anteriormente. Os novos algoritmos substituem

nossos algoritmos da primeira geração, chamada Zeta 1. Por um curto período de tempo, nós chamávamos os novos algoritmos de "Representações Distribuídas de Densidade Fixa", ou "FDR", mas já não estamos mais usando essa terminologia. Nós chamamos os novos algoritmos de Algoritmos de Aprendizagem Cortical HTM, ou às vezes simplesmente de Algoritmos de Aprendizagem HTM.

Nós encorajamos você a ler *Sobre a Inteligência*, escrito pelo co-fundador da Numenta Jeff Hawkins e por Sandra Blakeslee. Embora o livro não mencione HTM pelo nome, ele proporciona uma explicação não-técnica e de fácil leitura da teoria HTM e da neurociência por trás dela. No momento em que *Sobre a Inteligência* foi escrito, nós entendíamos os princípios básicos subjacentes à HTM mas não sabíamos como aplicar esses princípios através de algoritmos. Você pode pensar no presente documento como a continuação do trabalho iniciado em *Sobre a Inteligência*.

Sobre a Numenta

A Numenta, Inc. (www.numenta.com) foi fundada em 2005 para desenvolver tecnologia HTM tanto para uso comercial como para uso científico. Para atingir este objetivo, estamos documentando plenamente o nosso progresso e nossas descobertas. Nós também publicamos nosso software de uma forma que outras pessoas possam usar tanto para pesquisa como para desenvolvimento comercial. Nós estruturamos nosso software para estimular o surgimento de uma comunidade de desenvolvedores de aplicativos independentes. O uso de software e propriedade intelectual Numenta é livre para fins de pesquisa. Nós vamos gerar receita com a venda de suporte, licenciamento de software e licenciamento de propriedade intelectual para implantações comerciais. Nós vamos sempre procurar fazer os nossos desenvolvedores parceiros terem sucesso, bem como nós mesmos sermos bem sucedidos.

A Numenta possui sede em Redwood City, Califórnia. Ela é financiada com fundos privados.

Sobre os autores

Este documento é um esforço de colaboração entre os funcionários da Numenta. Os nomes dos autores principais de cada seção estão listados no histórico de revisão.

Histórico de revisão

Notamos na tabela abaixo as principais mudanças entre as versões. Pequenas alterações, tais como esclarecimentos ou pequenas alterações de formatação não são observadas.

Versão	Data	Modificações	Principais autores
0.1	09 Nov 2010	1. Prefácio, Capítulos 1,2,3,4 e Glossário: primeira versão	Jeff Hawkins, Subedei Ahmad, Donna Dubinsky
0.1.1	23 Nov 2010	1. Capítulo 1: a seção Regiões foi editada para esclarecer a terminologia, tais como	Hawkins e Dubinsky

		níveis, colunas e camadas 2. Apêndice A: primeira versão	Hawkins
0.2	10 Dez 2010	1. Capítulo 2: esclarecimentos diversos 2. Capítulo 4: referências de linha atualizadas; alterações do código nas linhas 37 e 39 3. Apêndice B: primeira versão	Hawkins Ahmad Hawkins

Capítulo 1: Visão Geral da HTM

A Memória Temporal Hierárquica (HTM) é uma tecnologia de aprendizagem de máquina que visa capturar as propriedades estruturais e algorítmicas do neocórtex.

O neocórtex é a base do pensamento inteligente no cérebro dos mamíferos. Visão, audição, tato, movimento, linguagem e planejamento de alto nível são todos realizados pelo neocórtex. Diante de tal conjunto diversificado de funções cognitivas, você espera que o neocórtex implemente igualmente um conjunto diverso de algoritmos neurais especializados. Não é esse o caso. O neocórtex exibe um padrão notavelmente uniforme de circuitos neurais. A evidência biológica sugere que o neocórtex implementa um conjunto de algoritmos para executar muitas funções distintas da inteligência.

A HTM fornece um quadro teórico para a compreensão do neocórtex e suas muitas capacidades. Até esta data temos implementado um pequeno subconjunto desse quadro teórico. Com o tempo, mais e mais da teoria será implementado. Hoje acreditamos que temos implementado um subconjunto suficiente do que é o neocórtex de forma a ter valor comercial e científico.

Programar HTMs é diferente de programar computadores tradicionais. Com os computadores de hoje, os programadores criam programas específicos para resolver problemas específicos. Em contrapartida, HTMs são treinadas através da exposição a um fluxo de dados sensoriais. As capacidades da HTM são determinadas principalmente pelo que foi exposto.

As HTMs podem ser vistas como um tipo de rede neural. Por definição, qualquer sistema que tenta modelar os detalhes da arquitetura do neocórtex é uma rede neural. No entanto, por si só, o termo "rede neural" não é muito útil, pois tem sido aplicado a uma grande variedade de sistemas. As HTMs modelam neurônios (chamados de células quando se refere a HTM), os quais estão organizados em colunas, em camadas, em regiões, e em uma hierarquia. Os detalhes são importantes, e neste respeito as HTMs são uma nova forma de rede neural.

Como o nome indica, a HTM é fundamentalmente um sistema baseado em memória. As redes HTM são treinadas em lotes de dados que variam no tempo, além de contar com o armazenamento de um grande conjunto de padrões e seqüências. A forma como os dados são armazenados e acessados é logicamente diferente do modelo padrão utilizado por programadores de hoje. A memória de computador clássica tem uma organização estável e não tem uma inerente noção do tempo. Um programador pode implementar qualquer tipo de estrutura e organização de dados em cima da memória estável do computador. Eles têm controle sobre como e onde a informação é armazenada. Em contrapartida, a memória HTM é mais restritiva. A memória HTM tem uma organização hierárquica e é inerentemente baseada no tempo. A informação é sempre armazenada em uma forma distribuída. Um usuário de uma HTM especifica o tamanho da hierarquia e com que será treinado o sistema, mas é a HTM que controla onde e como as informações serão armazenadas.

Embora as redes HTM sejam substancialmente diferentes da computação clássica, podemos utilizar os computadores de uso geral para modelá-las, contanto que incorporem as principais funções das representações distribuídas no espaço, tempo e hierarquia (descrito em detalhe mais tarde). Acreditamos que ao longo do tempo, será criado hardware especializado para gerar redes proposadamente construídas em HTM.

Neste documento, nós muitas vezes ilustramos as propriedades e os princípios HTM usando exemplos tirados da visão, tato, audição, linguagem e comportamento humano. Tais exemplos são úteis porque são intuitivos e facilmente compreendidos. No entanto, é importante manter em mente que as capacidades HTM são gerais. Elas podem ser facilmente expostas à fluxos de entradas sensoriais não-humanos, como radar e infravermelho, ou à fluxos de entradas puramente informativos, tais como dados do mercado financeiro, dados de clima, padrões de tráfego da Web, ou texto. As HTMs são máquinas de aprendizagem e predição que podem ser aplicadas a vários tipos de problemas.

Princípios da HTM

Nesta seção, iremos cobrir alguns dos princípios fundamentais da HTM: por que a organização hierárquica é importante, como as regiões HTM são estruturadas, porque os dados são armazenados como representações distribuídas esparsas, e porque informação baseada no tempo é indispensável.

Hierarquia

Uma rede HTM é composta por regiões organizadas em uma hierarquia. A região é a principal unidade de memória e predição de uma HTM, e será discutida em detalhes na próxima seção. Normalmente, cada região HTM representa um nível na hierarquia. Conforme você sobe na hierarquia, há sempre convergência, vários elementos convergem de uma região filha para um elemento em uma região pai. No entanto, devido às conexões de realimentação, a informação também diverge assim que você desce na hierarquia. (Uma "região" e um "nível" são quase sinônimos. Nós usamos o termo "região", quando descrevemos a função interna de uma região, enquanto que usamos o termo "nível" quando se refere especificamente ao papel da região na hierarquia).

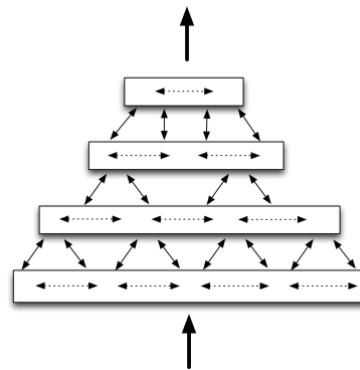


Figura 1.1: Diagrama simplificado de quatro regiões HTM organizadas em uma hierarquia de quatro níveis, comunicando informação dentro de níveis, entre os níveis, e de/para fora da hierarquia

É possível combinar múltiplas redes HTM. Este tipo de estrutura faz sentido se você tiver dados de mais de uma fonte ou sensor. Por exemplo, uma rede pode processar informação auditiva e uma outra rede pode processar informação visual. Há uma convergência dentro de cada rede distinta, com os ramos distintos convergindo somente no topo.

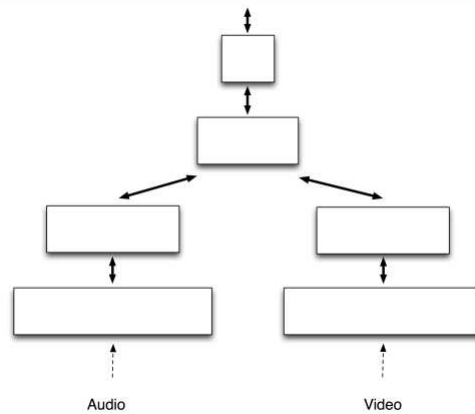


Figura 1.2: Redes convergentes de diferentes sensores

O benefício da organização hierárquica é a eficiência. Reduz significativamente o tempo de treinamento e uso da memória, porque padrões aprendidos em cada nível da hierarquia são reutilizados quando combinados de forma inédita em níveis mais altos. Para uma ilustração, vamos considerar a visão. No nível mais baixo da hierarquia, o cérebro armazena informações sobre seções pequenas do campo visual, tais como bordas e cantos. Uma aresta é um componente fundamental de muitos objetos no mundo. Esses padrões de baixo nível são recombinados para padrões de médio nível na forma de componentes mais complexos, tais como curvas e texturas. Um arco pode ser a ponta de uma orelha, o topo de um volante ou a borda de um copo de café. Estes padrões de médio nível são ainda combinados para representar características de objetos de alto nível, tais como cabeças, carros ou casas. Para aprender um novo objeto de alto nível você não tem que reaprender seus componentes.

Um outro exemplo, considere que quando você aprende uma nova palavra, você não precisa reaprender letras, sílabas ou fonemas.

O compartilhamento de representações em uma hierarquia também leva à generalização do comportamento esperado. Quando você vê um novo animal, se você vê uma boca e os dentes você vai predizer que o animal come com a boca e que poderia mordê-lo. A hierarquia permite que um novo objeto no mundo herde as propriedades conhecidas dos seus sub-componentes.

Quanto um único nível em uma hierarquia HTM pode aprender? Ou, dito de outra forma, quantos níveis na hierarquia são necessários? Há um equilíbrio entre a quantidade de memória alocada para cada nível e quantos níveis são necessários. Felizmente, HTMs automaticamente aprendem as melhores representações possíveis em cada nível dado as estatísticas da entrada e da quantidade de recursos alocados. Se você alocar mais memória para um nível, esse nível vai formar representações que são maiores e mais complexas, que por sua vez significa que menos níveis hierárquicos podem ser necessários. Se você alocar menos memória, um nível fará representações que são menores e mais simples, o que significa que mais níveis hierárquicos podem ser necessários.

Até este ponto, temos descrito os problemas difíceis, tais como inferência de visão ("inferência" é semelhante à reconhecimento de padrões). Mas muitos problemas valiosos são mais simples do que a visão, e uma única região HTM pode ser suficiente. Por exemplo, nós aplicamos uma HTM para predizer quando uma pessoa visitará um site clicando em Avançar. Esse problema envolve alimentar a rede HTM com fluxos de dados de cliques web. Neste problema, há pouca ou nenhuma hierarquia espacial, a solução de maior necessidade é descobrir as estatísticas temporais, ou seja, predizer onde o usuário deverá clicar em Avançar por reconhecer padrões

típicos do usuário. Os algoritmos de aprendizagem temporal em HTMs são ideais para tais problemas.

Em resumo, hierarquias reduzem o tempo de treinamento, reduzem o uso da memória, e introduzem uma forma de generalização. No entanto, muitos problemas mais simples de predição podem ser resolvidos com uma única região HTM.

Regiões

A noção de regiões conectadas em uma hierarquia vem da biologia. O neocórtex é uma grande folha de tecido neural com aproximadamente 2mm de espessura. Biólogos dividem o neocórtex em diferentes áreas ou "regiões" baseando-se principalmente em como regiões estão conectadas umas com as outras. Algumas regiões recebem entrada direta a partir dos sentidos e outras regiões recebem entrada somente depois que ela passou por várias outras regiões. É a conectividade de região-para-região, que define a hierarquia.

Todas as regiões neocorticais são semelhantes em seus detalhes. Elas variam em tamanho e de onde elas estão na hierarquia, mas fora isso, elas são semelhantes. Se você pegar uma fatia de 2mm de espessura de uma região neocortical, você verá seis camadas, cinco camadas de células e uma camada não-celular (há algumas exceções, mas esta é a regra geral). Cada camada de uma região neocortical possui muitas células interconectadas dispostas em colunas. Regiões HTM também são compostas por uma camada de células altamente interconectadas dispostas em colunas. A "Camada 3" no neocórtex é uma das principais camadas de alimentação dos neurônios. As células em uma região HTM são praticamente equivalentes aos neurônios na camada 3 de uma região do neocórtex.

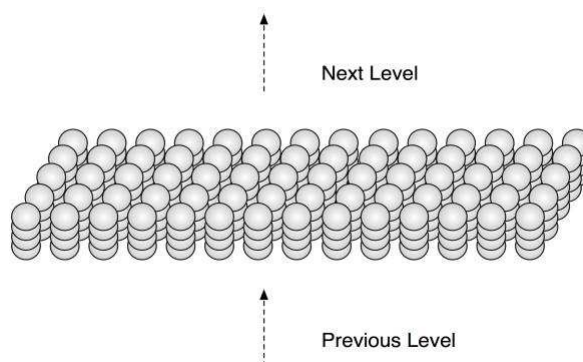


Figura 1.3: Uma seção de uma região HTM. Regiões HTM são compostas por muitas células. As células são organizadas em uma matriz bidimensional de colunas. Esta figura mostra uma pequena parte de uma região HTM com quatro células por coluna. Cada coluna se conecta a um subconjunto da entrada e cada célula se conecta a outras células na região (conexões não mostradas). Note que esta região HTM, incluindo a sua estrutura colunar, é equivalente a uma camada de neurônios em uma região neocortical.

Apesar de uma região HTM ser equivalente a apenas uma parte de uma região neocortical, ela pode fazer inferência e predição em fluxos de dados complexos e, portanto, pode ser útil em muitos problemas.

Representações Distribuídas Esparsas

Embora os neurônios no neocórtex estejam fortemente interconectados, os neurônios inibitórios garantem que apenas uma pequena porcentagem dos neurônios fiquem ativos ao mesmo tempo. Assim, a informação no cérebro é sempre representada por uma pequena porcentagem de

neurônios ativos dentro de uma grande população de neurônios. Este tipo de codificação é chamado de "representação distribuída esparsa". "Esparsa" significa que apenas uma pequena porcentagem de neurônios estão ativos ao mesmo tempo. "Distribuídas" significa que as ativações de muitos neurônios são necessárias a fim de representar alguma coisa. Um único neurônio ativo transmite algum significado, mas deve ser interpretado dentro do contexto de uma população de neurônios para transmitir o significado completo.

Regiões HTM também usam representações distribuídas esparsas. Na verdade, os mecanismos de memória dentro de uma região HTM são dependentes da utilização de representações distribuídas esparsas, e não de outra forma. A entrada para uma região HTM é sempre uma representação distribuída, mas nem sempre é esparsa, então a primeira coisa que uma região HTM faz é converter sua entrada em uma representação distribuída esparsa.

Por exemplo, uma região pode receber 20.000 bits de entrada. A porcentagem de bits de entrada que são "1" e "0" pode variar significativamente ao longo do tempo. Uma hora pode haver 5.000 bits "1" e outra hora pode haver 9.000 bits "1". A região HTM poderia converter esta entrada em uma representação interna de 10.000 bits, dos quais 2%, ou 200, estão ativos ao mesmo tempo, independentemente de quantos dos bits de entrada forem "1". Como a entrada para a região HTM varia ao longo do tempo, a representação interna também vai mudar, mas sempre haverá cerca de 200 bits em 10.000 ativos.

Pode parecer que esse processo gera uma grande perda de informações pois o número de possíveis padrões de entrada é muito maior que o número de possíveis representações na região. No entanto, ambos os números são incrivelmente grandes. As entradas reais vistas por uma região será uma fração minúscula de todas as entradas possíveis. Mais tarde iremos descrever como uma região cria uma representação esparsa de sua entrada. A perda de informações teóricas não terá um efeito prático.

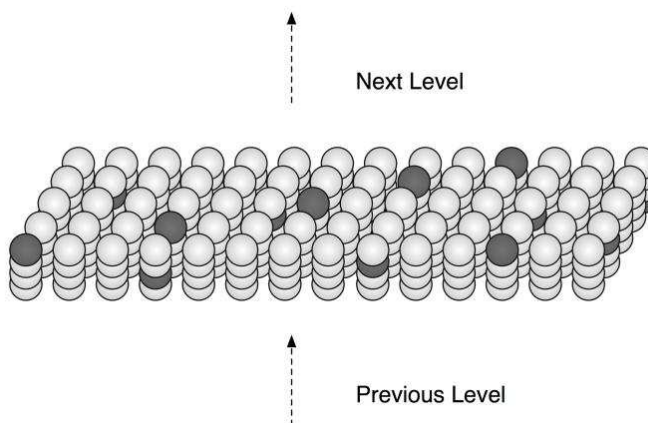


Figura 1.4: Uma região HTM mostrando a ativação de células distribuídas esparsas

Representações distribuídas esparsas têm várias propriedades desejáveis e são essenciais para o funcionamento das HTMs. Falaremos sobre elas novamente mais tarde.

O papel do tempo

O tempo desempenha um papel crucial na aprendizagem, inferência e predição.

Vamos começar com a inferência. Sem o uso do tempo, podemos inferir quase nada dos nossos sentidos tátil e auditivo. Por exemplo, se você estiver de olhos fechados e alguém colocar uma maçã em sua mão, você poderá identificar o que é só depois de manipulá-la por um segundo ou mais. A medida que você move os dedos sobre a maçã, embora a informação tátil esteja mudando constantemente, o próprio objeto - a maçã, bem como a sua percepção de alto nível para "maçã" - permanece constante. No entanto, se uma maçã foi colocada na sua palma da mão estendida, e você não foi autorizado a passar a mão ou os dedos, você terá grande dificuldade de identificá-la como uma maçã em vez de um limão.

O mesmo vale para a audição. Um som estático transmite pouco significado. Uma palavra como "maçã", ou os sons de mastigação de alguém mordendo uma maçã, só podem ser reconhecidos em dezenas ou centenas de mudanças rápidas e sequenciais ao longo do tempo do espectro sonoro.

A visão, em contraste, é um caso misto. Ao contrário do tato e da audição, os seres humanos são capazes de reconhecer as imagens quando elas são exibidas na sua frente tão rápido que nem dá chance para os olhos se moverem. Assim, a inferência visual nem sempre exige entradas variáveis ao longo do tempo. No entanto, durante a visão normal, estamos constantemente movendo os nossos olhos, cabeças e corpos, e objetos no mundo se movem ao nosso redor também. Nossa capacidade de inferir com base na rápida exposição visual é um caso especial possibilitado pelas propriedades estatísticas da visão e anos de treinamento. O caso geral para a visão, audição e tato é que a inferência requer entradas variáveis ao longo do tempo.

Tendo coberto o caso geral da inferência, e no caso particular da inferência da visão em imagens estáticas, vamos dar uma olhada na aprendizagem. A fim de aprender, todos os sistemas HTM devem ser expostos a entradas variáveis ao longo do tempo durante o treinamento. Mesmo na visão, onde a inferência estática às vezes é possível, temos de ver mudar as imagens dos objetos para saber com o que um objeto se parece. Por exemplo, imagine um cão que esteja correndo em sua direção. A cada instante no tempo o cão provoca um padrão de atividade na retina do seu olho. Você percebe esses padrões como diferentes visões do mesmo cão, mas matematicamente os padrões são totalmente diferentes. O cérebro aprende que esses diferentes padrões significam a mesma coisa, observando-os em sequência. O tempo é o "supervisor", ensinando-lhe que os padrões espaciais caminham juntos.

Note que não é suficiente para a entrada sensorial apenas mudar. Uma sucessão de padrões sensoriais independentes só levaria à confusão. As entradas variáveis ao longo do tempo devem vir de uma fonte comum no mundo. Note também que, embora usemos sentidos humanos como exemplo, o caso geral aplica-se aos sentidos não-humanos também. Se queremos treinar uma HTM para reconhecer padrões de sensores de temperatura, vibração e ruído de uma usina, a HTM precisa ser treinada com dados dos sensores variando ao longo do tempo.

Normalmente, uma rede HTM precisa ser treinada com grandes quantidades de dados. Você aprendeu a identificar cães por ver muitos exemplos de muitas raças de cães, não apenas uma visão única de um único cão. A função dos algoritmos HTM é aprender as sequências temporais de um fluxo de dados de entrada, ou seja, construir um modelo dos padrões que seguem outros padrões. Esta tarefa é difícil porque ele pode não saber quando as sequências iniciam ou terminam, pode haver sobreposição de sequências que ocorrem ao mesmo tempo, a aprendizagem tem que ocorrer de forma contínua, e a aprendizagem tem que ocorrer na presença de ruído.

Aprender e reconhecer seqüências é a base da formação de predições. Uma vez que uma HTM aprende quais padrões são propensos a seguir outros padrões, ela pode predizer o próximo (ou próximos) padrão provável dado a entrada atual e as últimas entradas passadas. A predição será abordada em mais detalhes mais tarde.

Vamos agora voltar para as quatro funções básicas da HTM: aprendizagem, inferência, predição e comportamento. Cada região HTM executa as três primeiras funções: aprendizagem, inferência e predição. O comportamento, entretanto, é diferente. Nós sabemos da biologia que a maioria das regiões neocorticais têm um papel na criação do comportamento, mas nós não acreditamos que ele seja essencial para muitas aplicações interessantes. Portanto, nós não incluímos o comportamento na nossa implementação atual da HTM. Citamos aqui para completar.

Aprendizagem

Uma região HTM aprende sobre o seu mundo, encontrando padrões e seqüências de padrões em dados sensoriais. A região não "sabe" o que representam suas entradas; ela trabalha em uma esfera puramente estatística. Ele procura por combinações de bits de entrada, que ocorrem frequentemente em conjunto, que chamamos de padrões espaciais. Em seguida, ela repara em como esses padrões espaciais aparecem em seqüência ao longo do tempo, os quais chamamos de padrões ou seqüências temporais.

Se a entrada para a região representa sensores ambientais em um edifício, a região poderá descobrir que certas combinações de temperatura e umidade no lado norte do edifício ocorrem com freqüência e que combinações diferentes ocorrem no lado sul do edifício. Em seguida, ele pode aprender que as seqüências dessas combinações ocorrem a cada dia que passa.

Se a entrada para uma região representada informações relacionadas às compras dentro de uma loja, a região HTM pode descobrir que certos tipos de artigos são comprados nos fins de semana, ou que quando o tempo está frio certas variações de preços são favorecidas durante a noite. Em seguida, ela pode aprender que diferentes indivíduos seguem padrões sequenciais semelhantes em suas compras.

Uma única região HTM tem capacidade de aprendizagem limitada. Uma região ajusta automaticamente o que aprende com base na quantidade de memória que tem e na complexidade das informações que recebe. Os padrões espaciais aprendidos por uma região se tornarão necessariamente mais simples, se a memória alocada para uma região for reduzida. Ou os padrões espaciais aprendidos podem tornar-se mais complexos, se a memória alocada for aumentada. Se os padrões espaciais aprendidos em uma região forem simples, então, uma hierarquia de regiões pode ser necessária para compreender imagens complexas. Vemos essa característica no sistema de visão humano, onde a região neocortical que recebe a entrada da retina aprende padrões espaciais de pequenas partes do espaço visual. Só depois de vários níveis de hierarquia que padrões espaciais se combinam e representam a maior parte ou a totalidade do espaço visual.

Como um sistema biológico, os algoritmos de aprendizagem em uma região HTM são capazes de "aprendizagem on-line", ou seja, aprender continuamente a partir de cada nova entrada. Não há necessidade de uma fase de aprendizagem isolada de uma fase de inferência, embora a inferência melhore depois de aprender mais. A medida que os padrões de entrada mudarem, a região HTM também irá mudar gradualmente.

Após o treinamento inicial, uma HTM pode continuar a aprender ou, alternativamente, a aprendizagem pode ser desativada após a fase de treinamento. Outra opção é desligar a aprendizagem só nos níveis mais baixos da hierarquia, mas continuar a aprender com os níveis mais elevados. Uma vez que uma HTM tenha aprendido a estrutura estatística básica de seu mundo, aprendizagens mais novas só ocorrerão nos níveis superiores da hierarquia. Se uma HTM é exposta a novos padrões que têm estrutura de baixo nível inéditas, vai levar mais tempo para a HTM aprender estes novos padrões. Vemos essa característica nos seres humanos. Aprender palavras novas em uma língua que você já sabe é relativamente fácil. No entanto, se você tentar aprender novas palavras de uma língua estrangeira com sons não muito familiares, você vai encontrar muito mais dificuldade porque você ainda não conhece os sons de baixo nível.

De fato, descobrir padrões é um recurso potencialmente valioso. Compreender os padrões de alto nível, as flutuações do mercado, doenças, clima, produção, fabricação ou falhas de sistemas complexos, tais como redes de energia, é importante em si mesmo. Mesmo assim, os padrões de aprendizagem espacial e temporal é sobretudo um precursor para a inferência e a predição.

Inferência

Depois de uma HTM aprender os padrões no seu mundo, ela pode fazer inferências em cima de novas entradas. Quando uma HTM recebe a entrada, ela a compara com os padrões espaciais e temporais aprendidos anteriormente. Combinações bem sucedidas de novas entradas para seqüências previamente armazenadas é a essência da inferência e da combinação de padrões.

Pense em como você reconhece uma melodia. Ouvir a primeira nota de uma melodia diz-lhe pouco. A segunda nota reduz significativamente as possibilidades, mas ainda pode não ser suficiente. Normalmente, leva-se três, quatro ou mais notas antes de você reconhecer a melodia. A inferência em uma região HTM é semelhante. Ela está constantemente fazendo uma busca em um fluxo de entradas e combinando-as às seqüências previamente aprendidas. Uma região HTM pode encontrar combinações desde o início das seqüências, mas geralmente ela é mais fluída, semelhante ao modo como você pode reconhecer uma melodia a partir de qualquer lugar. Como as regiões HTM usam representações distribuídas, o uso de memória de seqüências e inferência de uma região é mais complicado do que o exemplo da melodia sugere, mas o exemplo dá uma ideia de como ele funciona.

Pode não ser imediatamente evidente, mas toda a experiência sensorial que você já teve foi nova, mas você facilmente encontra padrões familiares nesta nova entrada. Por exemplo, você pode entender a palavra "café" falada praticamente por qualquer pessoa, não importa se são velhos ou jovens, homens ou mulheres, se estão falando depressa ou devagar, ou se tem um sotaque forte. Mesmo se você tivesse a mesma pessoa dizendo a mesma palavra "café" uma centena de vezes, o som nunca iria estimular a sua cóclea (receptores auditivos), exatamente da mesma forma duas vezes.

Uma região HTM enfrenta o mesmo problema que o seu cérebro: as entradas não podem se repetir exatamente. Por conseguinte, tal como o seu cérebro, uma região HTM deve manipular a nova entrada durante a inferência e o treinamento. Um modo de uma região HTM lidar com a nova entrada é através do uso de representações distribuídas esparsas. A propriedade chave das representações distribuídas esparsas é que você só precisa combinar uma parte do padrão de forma a assegurar que a combinação seja significativa.

Predição

Cada região de uma HTM armazena seqüências de padrões. Ao combinar as seqüências armazenadas com a entrada atual, uma região forma uma predição sobre quais entradas provavelmente chegarão em seguida. Regiões HTM realmente armazenam transições entre representações distribuídas esparsas. Em alguns casos, as transições podem ser parecidas à uma seqüência linear, como as notas de uma melodia, mas no caso geral muitas possíveis entradas futuras podem ser preditas ao mesmo tempo. Uma região HTM fará predições diferentes com base no contexto que pode se estender de volta no tempo. A maior parte da memória em uma HTM é dedicada à memória de seqüência, ou ao armazenamento de transições entre os padrões espaciais.

A seguir estão algumas características-chave da predição HTM.

1) A predição é contínua.

Sem estar consciente disso, você está constantemente predizendo. HTMs fazem o mesmo. Ao ouvir uma música, você está predizendo a próxima nota. Ao descer as escadas, você está predizendo quando seu pé irá tocar o próximo degrau. Ao assistir a um lançamento de uma bola de beisebol, você está predizendo que a bola irá chegar perto do taco. Em uma região HTM, predição e inferência são quase a mesma coisa. Predição não é uma etapa separada, mas parte integrante da forma como funciona uma região HTM.

2) A predição ocorre em todas as regiões em todos os níveis da hierarquia.

Se você tem uma hierarquia de regiões HTM, a predição irá ocorrer em cada nível. As regiões irão fazer predições em cima dos padrões que elas aprenderam. No exemplo da linguagem, as regiões de baixo nível podem prever possíveis fonemas que virão, e as regiões de nível superior podem prever as palavras ou frases.

3) As predições são sensíveis ao contexto.

As predições são baseadas no que ocorreu no passado, bem como no que está ocorrendo agora. Assim, uma entrada irá produzir predições diferentes com base no contexto anterior. Uma região HTM aprende a utilizar o contexto anterior o quanto for necessário, e pode manter o contexto tanto em cima de intervalos curtos de tempo como em cima de longos. Esta capacidade é conhecida como memória de "ordem variável". Por exemplo, pense em um discurso memorizado, como o Discurso de Gettysburg. Para prever a próxima palavra, sabendo que somente a palavra atual raramente é suficiente; a palavra "e" é seguida por "sete" e depois por "dedicado" só na primeira frase. Às vezes, um pouco do contexto ajudará a predição; sabendo "quatro pontos e" poderia ajudar a prever o "sete". Outras vezes, há frases repetitivas, e seria preciso usar o contexto de um período de tempo muito mais longo para saber aonde você está no discurso e, portanto, o que vem depois.

4) A predição conduz à estabilidade.

A saída de uma região é a sua predição. Uma das propriedades das HTMs é que as saídas das regiões tornam-se mais estáveis - isto é, de mudança mais lenta e de maior duração - quanto mais altas elas estiverem na hierarquia. Esta propriedade resulta da forma como uma região prediz. Uma região não prediz apenas o que irá acontecer imediatamente em seguida. Se puder, ela irá prever vários intervalos à frente no tempo. Vamos dizer que uma região possa prever cinco intervalos à frente. Quando chega uma nova entrada, o recente intervalo predito muda, mas os quatro intervalos previamente preditos não. Consequentemente, mesmo que cada nova entrada seja completamente diferente, apenas uma parte da saída está mudando, fazendo saídas mais estáveis do que as entradas. Esta característica reflete a nossa experiência do mundo real,

onde conceitos de alto nível - como o nome de uma canção - mudam mais lentamente do que os conceitos de baixo nível - as notas reais da música.

5) A predição nos diz se uma nova entrada é esperada ou inesperada.

Cada região HTM é um detector de novidade. Como cada região prediz o que irá acontecer em seguida, ela "sabe" quando algo inesperado acontece. HTMs podem prever simultaneamente muitas possíveis entradas que virão em seguida, não apenas uma. Por isso, ela pode não ser capaz de prever com exatidão o que irá acontecer, mas se a próxima entrada não corresponder a nenhuma das previsões a região HTM saberá que uma anomalia ocorreu.

6) A predição ajuda a tornar o sistema mais robusto ao ruído.

Quando uma HTM prediz o que é susceptível de acontecer em seguida, a predição pode influenciar o sistema para inferir o que foi previsto. Por exemplo, se uma HTM estivesse processando linguagem falada, ela predizeria quais sons, palavras e idéias tenderiam a ser proferidos em seguida. Esta predição ajuda o sistema a preencher dados faltantes. Se um som ambíguo chega, a HTM irá interpretar o som baseado no que ela está esperando, contribuindo assim para a inferência, mesmo na presença de ruído.

Em uma região HTM, memória de seqüências, inferência e predição estão intimamente integrados. Elas são as funções básicas de uma região.

Comportamento

O nosso comportamento influencia o que percebemos. Enquanto nós movemos nossos olhos, nossa retina recebe estímulos sensoriais variáveis. Mover os nossos membros e dedos faz com que sensações variadas de tato cheguem ao cérebro. Quase todas as nossas ações mudam o que sentimos. Entrada sensorial e comportamento motor estão intimamente entrelaçados.

Durante décadas, a visão predominante era de que uma única região no neocórtex, a região motora primária, que originava os comandos motores no neocórtex. Com o tempo, descobriu-se que a maior parte ou todas as regiões no neocórtex têm uma saída motora, até mesmo as regiões de baixo nível sensorial. Parece que todas as regiões corticais integram funções sensoriais e motoras.

Esperamos que uma saída motora possa ser adicionada à cada região HTM dentro do modelo atualmente existente de forma que sejam gerados comandos motores de um modo semelhante a como se faz previsões. No entanto, todas as implementações de HTMs até a data atual foram puramente sensoriais, sem um componente motor.

Progresso em direção à implementação da HTM

Temos feito progressos substanciais transformando o modelo teórico da HTM em uma tecnologia prática. Temos implementado e testado várias versões dos algoritmos de aprendizagem cortical HTM e temos encontrado a arquitetura básica que se encaixa bem. À medida que testamos os algoritmos em novos conjuntos de dados, vamos aprimorando os mesmos e adicionando peças que faltam. Iremos atualizar este documento a medida formos fazendo isto. Os próximos três capítulos descrevem o estado atual dos algoritmos.

Há muitos componentes da teoria que ainda não estão implementados, incluindo em especial, a realimentação entre as regiões, cronometragem específica, e integração comportamento/sensorial-motor. Estes componentes em falta devem ser inseridos no modelo já criado.

Capítulo 2: Algoritmos de Aprendizagem Cortical HTM

Este capítulo descreve os algoritmos de aprendizagem funcionando dentro de uma região HTM. Os capítulos 3 e 4 descrevem a aplicação dos algoritmos de aprendizagem utilizando pseudocódigo, visto que este capítulo é mais conceitual.

Terminologia

Antes de começar, uma nota sobre a terminologia pode ser útil. Usamos a linguagem da neurociência na descrição dos algoritmos de aprendizagem HTM. Termos tais como células, sinapses, sinapses potenciais, segmentos dendríticos, e colunas são usados por toda parte. Esta terminologia é lógica, já que os algoritmos de aprendizagem foram em grande parte, derivados de combinações de detalhes da neurociência com necessidades teóricas. No entanto, no processo de implementação dos algoritmos nós fomos confrontados com problemas de desempenho e, portanto, uma vez que nós sentíamos que tínhamos entendido como algo funcionava, procurávamos maneiras de aumentar a velocidade de processamento. Isso muitas vezes envolveu desviar-se de uma adesão estrita aos detalhes biológicos, enquanto pudéssemos obter os mesmos resultados. Se você é novo na neurociência isso não será um problema. No entanto, se você estiver familiarizado com os termos da neurociência, você pôde ficar confuso com o uso de termos fugindo de sua expectativa. Os apêndices sobre biologia discutem em detalhes as diferenças e semelhanças entre os algoritmos de aprendizagem HTM e seus equivalentes neurobiológicos. Aqui vamos citar alguns dos desvios que podem causar uma maior confusão.

Estados de célula

Células HTM possuem três estados de saída, ativo a partir da entrada de alimentação, ativo a partir da entrada lateral (o que representa uma predição) e inativo. O primeiro estado de saída corresponde a uma curta rajada de potenciais de ação em um neurônio. O segundo estado de saída corresponde a uma taxa mais lenta, constante de potenciais de ação em um neurônio. Nós não encontramos necessidade de modelar potenciais individuais de ação ou mesmo taxas escalares de atividade para além dos dois estados ativos. O uso de representações distribuídas parece superar a necessidade de modelar taxas de atividade escalar nas células.

Segmentos dendríticos

As células HTM têm um modelo relativamente realista (e, portanto, complexo) do modelo dendrítico. Em teoria, cada célula HTM tem um segmento dendrítico proximal e uma ou duas dúzias de segmentos dendríticos distais. O segmento dendrítico proximal recebe entrada de alimentação e os segmentos dendríticos distais recebem entrada lateral a partir de células próximas. Uma classe de células inibitórias força todas as células de uma coluna a responderem à entrada de alimentação semelhante. Para simplificar, removemos o segmento dendrítico proximal de cada célula e o substituímos por um único segmento dendrítico compartilhado por coluna de células. A função agrupador espacial (descrita abaixo) opera no segmento dendrítico compartilhado, a nível de colunas. A função agrupador temporal opera nos segmentos dendríticos distais, a nível de células individuais nas colunas. Esta simplificação atinge a mesma funcionalidade, mesmo que na biologia não exista nada equivalente a um segmento dendrítico anexado a uma coluna.

Sinapses

Sinapses HTM têm pesos binários. As sinapses biológicas têm diferentes pesos, mas elas também são parcialmente estocásticas, sugerindo que um neurônio biológico não pode invocar

pesos sinápticos precisos. O uso de representações distribuídas em HTMs somado ao nosso modelo de operação dendrítica nos permite atribuir pesos binários à sinapses HTM sem nenhum efeito negativo. Para modelar a formação e a não-formação de sinapses nós usamos dois conceitos complementares da neurociência que você pode não estar familiarizado. Um deles é o conceito de "sinapses potenciais". Isso representa todos os axônios que passam perto o suficiente de um segmento dendrítico ao ponto de poderem formar uma sinapse. O segundo é chamado de "permanência". Este é um valor escalar atribuído a cada sinapse potencial. A permanência de uma sinapse representa um intervalo de conexão entre um axônio e um dendrito. Biologicamente, a faixa iria de totalmente desconectada, para começando a formar uma sinapse mas ainda não conectada, depois para uma sinapse minimamente conectada, e finalmente para uma sinapse ampla e totalmente conectada. A permanência de uma sinapse é um valor escalar variando de 0,0 a 1,0. Aprender envolve incrementar e decrementar a permanência de uma sinapse. Quando a permanência de uma sinapse é acima de um limite, ela é conectada com um peso de "1". Quando está abaixo do limite, ela é desconectada com um peso de "0".

Visão Geral

Imagine que você seja uma região de uma HTM. Sua entrada consiste em milhares ou dezenas de milhares de bits. Estes bits de entrada podem representar dados sensoriais ou podem vir de outra região mais baixa na hierarquia. Eles estão ligando e desligando de forma complexa. O que você deveria fazer com essa entrada?

Nós já temos discutido a resposta em sua forma mais simples. Cada região HTM procura por padrões comuns em sua entrada e, em seguida, aprende seqüências desses padrões. Através da sua memória de seqüências, cada região faz predições. Essa descrição de alto nível faz isso parecer fácil, mas na realidade há muita coisa acontecendo. Vamos dividi-la um pouco mais nas três etapas seguintes:

- 1) Formação de uma representação distribuída esparsa da entrada
- 2) Formação de uma representação da entrada no contexto das entradas anteriores
- 3) Formação de uma predição com base na entrada atual no contexto das entradas anteriores

Vamos discutir cada uma dessas etapas em detalhes.

1) Formação de uma representação distribuída esparsa da entrada

Quando você imagina uma entrada para uma região, você pensa nela como um grande número de bits. Em um cérebro estes seriam os axônios dos neurônios. Em qualquer ponto no tempo alguns destes bits de entrada estarão ativos (valor 1) e outros estarão inativos (valor 0). A percentagem de bits de entrada que estarão ativos varia, digamos, de 0% a 60%. A primeira coisa que uma região HTM faz é converter esta entrada em uma nova representação que é esparsa. Por exemplo, a entrada pode ter 40% de seus bits "ligados", mas a nova representação ter apenas 2% dos seus bits "ligados".

Uma região HTM é logicamente constituída por um conjunto de *colunas*. Cada coluna é composta de uma ou mais *células*. As colunas podem ser logicamente organizadas em uma matriz 2D, mas isto não é uma exigência. Cada coluna em uma região é conectada a um único subconjunto de bits de entrada (normalmente sobreposto com outras colunas, mas nunca exatamente o mesmo subconjunto de bits de entrada). Como resultado, diferentes padrões de entrada resultam em diferentes níveis de ativação das colunas. As colunas com maior ativação inibem, ou desativam, as colunas com menor ativação. (A inibição ocorre dentro de um raio que

pode abranger desde o muito local até uma região inteira.) A representação esparsa da entrada é codificada através de quais colunas estão ativas e quais estão inativas após a inibição. A função da inibição se define em atingir uma porcentagem relativamente constante de colunas a estarem ativas, mesmo quando o número de bits de entrada que são ativos varia significativamente.

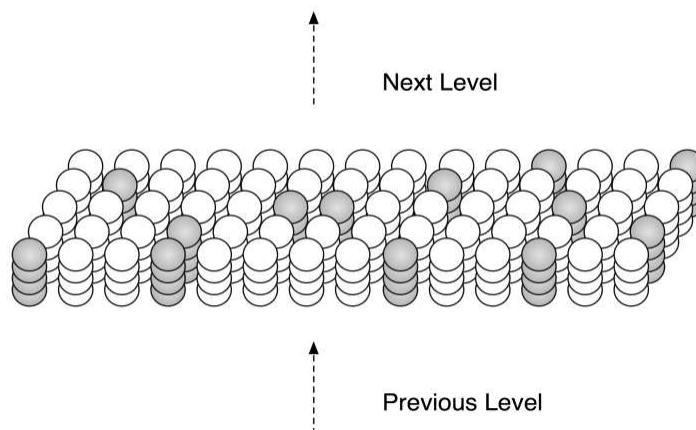


Figura 2.1: Uma região HTM consiste de colunas de células. Apenas uma pequena parte de uma região é mostrada.

Cada coluna de células recebe a ativação de um único subconjunto de entrada. Colunas com maior ativação inibem colunas com menor ativação. O resultado é uma representação distribuída esparsa da entrada. A figura mostra colunas ativas em cinza claro. (Quando não há um estado prévio, todas as células nas colunas ativas estarão ativas, como mostrado.)

Imagine agora que o padrão de entrada muda. Se apenas alguns bits de entrada mudam, algumas colunas receberão um pouco mais ou um pouco menos entradas no estado "ligado", mas o conjunto de colunas ativas provavelmente não vai mudar muita coisa. Assim semelhantes padrões de entrada (aqueles que têm um número significativo de bits ativos em comum) irão mapear um conjunto relativamente estável de colunas ativas. O quão estável a codificação será, dependerá muito de quais entradas cada coluna está conectada. Essas conexões são aprendidas através de um método descrito mais tarde.

Todos estes passos (aprender as conexões para cada coluna de um subconjunto das entradas, determinar o nível de entrada para cada coluna, e usar a inibição para selecionar um conjunto ativo de colunas esparsas) são referidos como "Agrupador Espacial". O termo significa que os padrões são "espacialmente" similares (o que significa que eles compartilham um grande número de bits ativos) e são "agrupados" (o que significa que eles estão agrupados em uma representação comum).

2) Formação de uma representação da entrada no contexto das entradas anteriores

A próxima função executada por uma região é a de converter a representação de colunas da entrada em uma nova representação, que inclui o estado ou contexto, do passado. A nova representação é formada pela ativação de um subconjunto de células em cada coluna, geralmente apenas uma célula por coluna (Figura 2.2).

Considere ouvir duas frases faladas, "I ate a pear" ("Eu comi uma pêra") e "I have eight pears" ("Eu tenho oito pêras"). As palavras "ate" e "eight" são homônimos; soam idênticas (no inglês). Podemos estar certos de que em algum ponto do cérebro há neurônios que respondem de forma idêntica para as palavras "ate" e "eight". Afinal, são sons idênticos que estão entrando no

ouvido. No entanto, nós também podemos estar certos de que em outro ponto do cérebro os neurônios que respondem a esta entrada são diferentes, em diferentes contextos. As representações para o som "ate" serão diferentes quando você ouvir "I ate" versus "I have eight". Imagine que você memorizou as duas frases "I ate a pear" e "I have eight pears". Escutar "I ate..." leva a uma predição diferente de "I have eight...". Deve haver diferentes representações internas depois de ouvir "I ate" e "I have eight".

Este princípio de codificação de uma entrada de forma diferente em contextos diferentes é uma característica universal de percepção e ação e é uma das funções mais importantes de uma região HTM. É difícil exagerar a importância dessa capacidade.

Cada coluna em uma região HTM consiste de várias células. Todas as células em uma coluna obtêm a mesma entrada de alimentação. Cada célula em uma coluna pode estar ativa ou inativa. Ao selecionar diferentes células ativas em cada coluna ativa, nós podemos representar a mesma entrada exata de forma diferente em diferentes contextos. Um exemplo específico pode ajudar. Digamos que cada coluna possua 4 células e a representação de cada entrada consista de 100 colunas ativas. Se apenas uma célula de cada coluna estiver ativa por vez, teremos 4^{100} maneiras de representar a mesma entrada exata. A mesma entrada resultará sempre nas mesmas 100 colunas sendo ativadas, mas em diferentes contextos diferentes células nessas colunas estarão ativas. Então nós podemos representar a mesma entrada em um número muito grande de contextos, mas como essas diferentes representações serão únicas? Quase todos os pares escolhidos aleatoriamente de 4^{100} possíveis padrões se sobrepõem em cerca de 25 células. Assim, duas representações de uma determinada entrada em diferentes contextos terão cerca de 25 células em comum e 75 células que são diferentes, tornando-as facilmente distinguíveis.

A regra geral utilizada por uma região HTM é a seguinte. Quando uma coluna se torna ativa, ela faz uma busca em todas as células na coluna. Se uma ou mais células na coluna já estão no estado preditivo, apenas estas células se tornam ativas. Se nenhuma célula na coluna está no estado preditivo, então todas as células se tornam ativas. Você pode pensar desta maneira, se um padrão de entrada é esperado, então o sistema confirma essa expectativa, ativando apenas as células no estado preditivo. Se o padrão de entrada é inesperado então o sistema ativa todas as células da coluna, como se dissesse "a entrada ocorreu inesperadamente assim todas as possíveis interpretações são válidas".

Se não houver um estado anterior e, portanto, sem contexto e predição, todas as células em uma coluna se tornarão ativas quando a coluna se torna ativa. Este cenário é semelhante ao ouvir a primeira nota de uma canção. Sem contexto, você normalmente não pode prever o que irá acontecer a seguir; todas as opções estão disponíveis. Se houver estado anterior, mas a entrada não corresponder ao que é esperado, todas as células da coluna ativa ficarão ativas. Esta determinação é feita em uma base de coluna à coluna assim a combinação ou a incompatibilidade de uma predição nunca será um evento de "tudo-ou-nada".

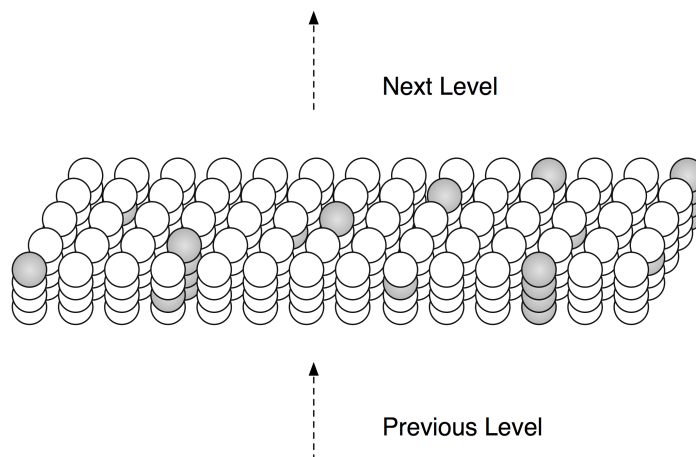


Figura 2.2: Ativando um subconjunto de células em cada coluna, uma região HTM pode representar a mesma entrada em muitos contextos diferentes. As colunas ativam apenas as células previstas. As colunas sem células previstas ativam todas as células na coluna. A figura mostra algumas colunas com uma célula ativa e algumas colunas com todas as células ativas.

Conforme mencionado na seção de terminologia acima, as células HTM podem estar em um dos três estados. Se uma célula está ativa devido à entrada de alimentação nós simplesmente usamos o termo "ativo". Se a célula está ativa devido às conexões laterais com outras células vizinhas, dizemos que ela está em "estado preditivo" (Figura 2.3).

3) Formação de uma predição com base na entrada no contexto das entradas anteriores

O passo final para a nossa região é fazer uma predição do que pode acontecer a seguir. A predição é baseada na representação formada na etapa 2), que inclui o contexto de todas as entradas anteriores.

Quando uma região faz uma predição ela ativa (para o estado preditivo) todas as células que provavelmente se tornarão ativas devido à entrada de alimentação futura. Devido as representações em uma região serem esparsas, várias predições podem ser feitas ao mesmo tempo. Por exemplo, se 2% das colunas está ativo devido a uma entrada, você poderia esperar que dez predições diferentes poderiam ser feitas resultando em 20% das colunas ter uma célula predita. Ou, vinte predições diferentes poderiam ser feitas resultando em 40% das colunas ter uma célula predita. Se cada coluna tivesse quatro células, com uma ativa de cada vez, então 10% das células estaria no estado preditivo.

Um capítulo futuro sobre representações distribuídas esparsas irá mostrar que apesar de diferentes predições serem mescladas juntas, uma região pode saber com grande certeza se uma entrada particular foi predita ou não.

Como uma região faz uma predição? Quando os padrões de entrada mudam ao longo do tempo, diferentes conjuntos de colunas e células se tornam ativos na sequência. Quando uma célula se torna ativa, ela faz conexões a um subconjunto de células vizinhas que estavam ativas imediatamente antes. Estas conexões podem ser formadas rapidamente ou lentamente dependendo da taxa de aprendizagem exigida pelo aplicativo. Posteriormente, tudo o que uma célula precisa fazer é procurar por atividade coincidente nestas conexões. Se as conexões tornam-se ativas, a célula pode esperar que ela possa se tornar ativa e logo entra em um estado preditivo. Assim, a ativação de alimentação de um conjunto de células levarão à ativação

preditiva de outros conjuntos de células que geralmente seguem. Pense nisso como o momento que você reconhece uma música e começa prever a nota seguinte.

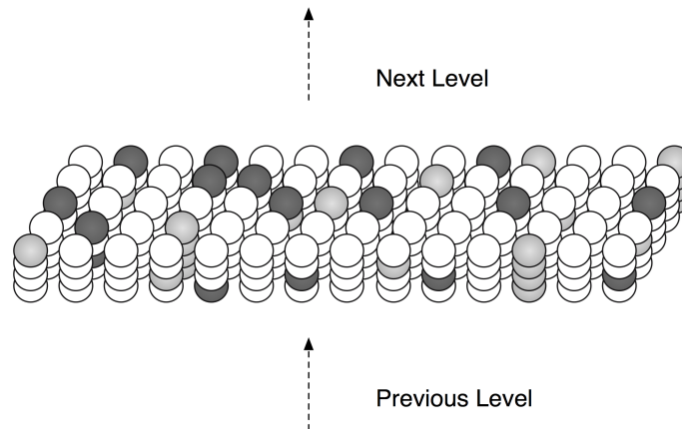


Figura 2.3: Em qualquer ponto no tempo, algumas células em uma região HTM serão ativas devido à entrada de alimentação (mostrado em cinza claro). As outras células que recebem a entrada lateral de células ativas estarão em um estado preditivo (mostrado em cinza escuro).

Em resumo, quando uma nova entrada chega, a mesma provoca um conjunto esparsos de colunas ativas. Uma ou mais células em cada coluna se tornam ativas, estas por sua vez, faz com que outras células entrem em um estado preditivo através de conexões aprendidas entre as células na região. As células ativadas por meio de conexões dentro da região constitui uma predição do que pode acontecer a seguir. Quando a próxima entrada de alimentação chega, ela escolhe um outro conjunto esparsos de colunas ativas. Se uma coluna recém-ativa é inesperada, o que significa que ela não foi predita por quaisquer células, ela vai ativar as células nas colunas. Se uma coluna recém-ativa tem uma ou mais células preditas, apenas estas células se tornarão ativas. A saída de uma região é a atividade de todas as células na região, incluindo as células ativas devido a uma entrada de alimentação e as células ativas no estado preditivo.

Como mencionado anteriormente, as predições não são apenas para o *próximo* intervalo no tempo. As predições de uma região HTM podem ser para vários intervalos de tempo no futuro. Usemos as melodias como exemplo, uma região HTM não prediz apenas a próxima nota de uma melodia, mas também pode prever as próximas quatro notas. Isso leva a uma propriedade desejável. A saída de uma região (a união de todas as células ativas e preditas em uma região) muda mais lentamente do que a entrada. Imagine que a região está predizendo as próximas quatro notas de uma melodia. Iremos representar a melodia pela seqüência de letras A, B, C, D, E, F, G. Depois de ouvir as duas primeiras notas, a região reconhece a seqüência e começa a prever. Ela prediz C, D, E, F. As células "B" já estão ativas, portanto, as células para B, C, D, E, F, estão todas em um dos dois estados ativos. Agora, a região ouve a próxima nota, "C". O conjunto de células ativas e preditivas representa agora "C, D, E, F, G". Note que o padrão de entrada mudou completamente passando de "B" para "C", mas apenas 20% das células mudaram.

Devido a saída de uma região HTM ser um vetor que representa a atividade de todas as células da região, a saída deste exemplo é cinco vezes mais estável que a entrada. Em um arranjo hierárquico de regiões, veremos um aumento na estabilidade temporal a medida que subirmos na hierarquia.

Nós usamos o termo "agrupador temporal" para descrever as duas etapas de adição de contexto na representação e na predição. Ao criar saídas de mudança lenta para as seqüências de padrões, nós estamos em essência "agrupando" juntos diferentes padrões que se seguem no tempo.

Agora vamos entrar em outro nível de detalhe. Começaremos com os conceitos que são compartilhados pelo agrupador espacial e pelo agrupador temporal. Em seguida, discutiremos os conceitos e detalhes exclusivos do agrupador espacial seguido pelos conceitos e detalhes exclusivos do agrupador temporal.

Conceitos compartilhados

A aprendizagem no agrupador espacial e no agrupador temporal são semelhantes. A aprendizagem em ambos os casos envolve o estabelecimento de conexões, ou sinapses, entre as células. O agrupador temporal aprende conexões entre as células na mesma região. O agrupador espacial aprende conexões de alimentação entre bits de entrada e colunas.

Pesos binários

Sinapses HTM têm apenas 0 ou 1 como efeito; seu "peso" é binário, uma propriedade diferente de muitos modelos de redes neurais que utilizam os valores de variáveis escalares na faixa de 0 a 1.

Permanência

As sinapses são formadas e desformadas constantemente durante a aprendizagem. Como mencionado anteriormente, vamos atribuir um valor escalar para cada sinapse (0,0 à 1,0) para indicar o quão permanente é a conexão. Quando uma conexão é reforçada, sua permanência é aumentada. Em outras condições, a permanência é diminuída. Quando a permanência for superior a um limite (por exemplo, 0,2), a sinapse é considerada estabelecida. Se a permanência for inferior ao limite, a sinapse não terá nenhum efeito.

Segmentos dendríticos

Sinapses se conectam a segmentos dendríticos. Existem dois tipos de segmentos dendríticos, o proximal e o distal.

- Um segmento dendrítico proximal forma sinapses com as entradas de alimentação. As sinapses ativas neste tipo de segmento são linearmente somadas para determinar a ativação de alimentação de uma coluna.

- Um segmento dendrítico distal forma sinapses com as células dentro da região. Cada célula possui diversos segmentos dendríticos distais. Se a soma das sinapses ativas em um segmento distal exceder um limite, então a célula associada se tornará ativa em um estado preditivo. Uma vez que existe múltiplos segmentos dendríticos distais por células, o estado preditivo de uma célula é a operação lógica OR de vários detectores de limite constituintes.

Sinapses Potenciais

Como mencionado anteriormente, cada segmento dendrítico tem uma lista de sinapses potenciais. Para todas as sinapses potenciais é dado um valor de permanência e elas podem tornar-se sinapses funcionais se os valores de sua permanência for superior a um limite.

Aprendizagem

Aprender envolve aumentar ou diminuir os valores de permanência das sinapses potenciais em um segmento dendrítico. As regras usadas para fazer sinapses mais ou menos permanentes são semelhantes às regras de aprendizagem "Hebbiana". Por exemplo, se uma célula pós-sináptica está ativa devido a um segmento dendrítico receber entradas acima do seu limite, então os valores de permanência das sinapses nesse segmento estão modificados. Sinapses que estão ativas, e portanto, contribuíram para a célula estar ativa, tem sua permanência aumentada. Sinapses que estão inativas, e portanto, não contribuíram, têm sua permanência diminuída. As condições exatas sob as quais os valores de permanência da sinapse são atualizados diferem no agrupador espacial e no agrupador temporal. Os detalhes estão descritos abaixo.

Agora iremos discutir conceitos específicos para as funções do agrupador espacial e do agrupador temporal.

Conceitos do agrupador espacial

A função mais importante do agrupador espacial é converter a entrada de uma região para um padrão esparsos. Essa função é importante porque o mecanismo usado para aprender seqüências e fazer predições necessita iniciar com padrões distribuídos esparsos.

Existem várias metas sobrepostas para o agrupador espacial, que determinam como o agrupador espacial opera e aprende.

1) Usar todas as colunas

Uma região HTM tem um número fixo de colunas que aprendem a representar padrões comuns na entrada. Um dos objetivos é garantir que todas as colunas aprendam a representar algo útil independentemente de quantas colunas você tenha. Nós não queremos colunas que nunca estão ativas. Para evitar que isto aconteça, acompanhamos quantas vezes uma coluna está ativa em relação às suas vizinhas. Se a atividade relativa de uma coluna é muito baixa, ela aumenta seu nível de atividade de entrada até que ela começa a fazer parte do conjunto vencedor de colunas. Em essência, todas as colunas estão competindo com as suas vizinhas para serem uma participante na representação de padrões de entrada. Se uma coluna não está muito ativa, ela vai se tornar mais agressiva. Quando isso acontecer, as outras colunas serão obrigadas a modificar as suas entradas e começar a representar padrões de entrada um pouco diferentes.

2) Manter a densidade desejada

Uma região necessita formar uma representação esparsa de suas entradas. Colunas com mais entradas inibem a suas vizinhas. Existe um raio de inibição que é proporcional ao tamanho dos campos receptivos das colunas (e portanto, pode variar desde o pequeno até o tamanho da região inteira). Dentro do raio de inibição, nós permitimos que apenas uma percentagem das colunas com mais entradas ativas sejam "vencedoras". As colunas remanescentes são desabilitadas. (Um "raio" de inibição implica um arranjo 2D de colunas, mas o conceito pode ser adaptado para outras topologias.)

3) Evitar padrões triviais

Queremos que todas as nossas colunas representem os padrões não-triviais na entrada. Esta meta pode ser alcançada mediante a fixação de um limite mínimo de participação para a coluna estar ativa. Por exemplo, se definimos o limite para 50, significará que uma coluna deverá ter no mínimo 50 sinapses ativas em seu segmento dendrítico para ela poder estar ativa, garantindo um certo nível de complexidade para o padrão que ela representa.

4) Evitar conexões extras

Se não formos cuidadosos, uma coluna poderá formar um grande número de sinapses válidas. Seria então uma resposta forte a diversos padrões de entrada independentes. Diferentes subconjuntos das sinapses iria responder a diferentes padrões. Para evitar esse problema, nós diminuimos o valor de permanência de qualquer sinapse que atualmente não está contribuindo para uma coluna vencedora. Por ter certeza que as sinapses não-contribuintes são suficientemente penalizadas, nós garantimos que uma coluna represente um número limitado de padrões de entrada, às vezes apenas um.

5) Auto ajustar campos receptivos

Os cérebros reais são muito "plásticos"; regiões do neocórtex podem aprender a representar coisas completamente diferentes em reação a várias mudanças. Se parte do neocórtex é danificada, outras partes irão se ajustar para representar o que a parte danificada usualmente representava. Se um órgão sensorial está danificado ou alterado, a parte associada do neocórtex irá se ajustar para representar outra coisa. O sistema é auto-ajustável.

Nós queremos que as nossas regiões HTM exibam a mesma flexibilidade. Se nós alocamos 10.000 colunas para uma região, ela deverá aprender a melhor forma para representar a entrada com 10.000 colunas. Se nós alocamos 20.000 colunas, ela deverá aprender a melhor forma para usar esse número. Se as estatísticas de entrada mudam, as colunas devem mudar para melhor representar a nova realidade. Em suma, o projeto de uma HTM deverá ser capaz de alocar os recursos para uma região e a região fará o melhor trabalho possível para poder representar a entrada com base nas colunas disponíveis e nas estatísticas de entrada. A regra geral é que com mais colunas em uma região, cada coluna representará padrões maiores e mais detalhados na entrada. Normalmente as colunas também estarão ativas com menos frequência, mas manteremos um nível de esparsidade relativamente constante.

Nenhuma nova regra de aprendizagem é necessária para alcançar essa meta altamente desejável. Estimulando colunas inativas, inibindo colunas vizinhas para manter a esparsidade constante, estabelecendo limites mínimos para a entrada, mantendo um grande conjunto de sinapses potenciais, e adicionando e esquecendo sinapses com base na sua contribuição, o conjunto de colunas irá se configurar dinamicamente para conseguir o efeito desejado.

Detalhes do agrupador espacial

Podemos agora passar por tudo o que a função de agrupamento espacial faz.

- 1) Começa com uma entrada que consiste em um número fixo de bits. Estes bits podem representar dados sensoriais ou podem vir de outra região mais baixa na hierarquia.
- 2) Atribui um número fixo de colunas para a região receber esta entrada. Cada coluna tem um segmento dendrítico associado. Cada segmento dendrítico tem um conjunto de sinapses potenciais que representa um subconjunto de bits de entrada. Cada sinapse potencial tem um valor de permanência. Com base nos valores de sua permanência, algumas das sinapses potenciais estarão válidas.
- 3) Para uma determinada entrada, determina quantas sinapses válidas em cada coluna são conectadas aos bits de entrada ativa.

4) O número de sinapses ativas é multiplicado por um fator de "estímulo" que é dinamicamente determinado pela frequência com que uma coluna está ativa em relação às suas vizinhas.

5) As colunas com as maiores ativações após o estímulo desabilitam todas as outras em uma percentagem fixa das colunas dentro de um raio de inibição. O raio de inibição é auto determinado dinamicamente pela extensão (ou "fan-out") de bits de entrada. Assim existe um conjunto ativo de colunas esparsas.

6) Para cada uma das colunas ativas, ajustamos os valores de permanência de todas as sinapses potenciais. Os valores de permanência das sinapses alinhadas com os bits de entrada ativa são aumentados. Os valores de permanência das sinapses alinhadas com bits de entrada inativa são diminuídos. As mudanças feitas nos valores de permanência podem mudar algumas sinapses de válidas para não-válidas, e vice-versa.

Conceitos do agrupador temporal

Lembre-se que o agrupador temporal aprende seqüências e faz predições. O método básico é que quando uma célula torna-se ativa, ela faz conexões com outras células que estavam ativas pouco antes. As células podem então predizer quando elas vão se tornar ativas, fazendo uma busca nas suas conexões. Se todas as células fizerem isso, coletivamente elas poderão armazenar e recuperar seqüências, e poderão predizer o que é susceptível de acontecer em seguida. Não há nenhuma central de armazenamento para uma seqüência de padrões; em vez disso, a memória é distribuída entre as células individuais. Devido a memória ser distribuída, o sistema é robusto ao ruído e ao erro. As células individuais podem falhar, geralmente com pouco ou nenhum efeito discernível.

É interessante notar algumas propriedades importantes das representações distribuídas esparsas que o agrupador temporal explora.

Suponha que temos uma região hipotética que sempre faz representações utilizando 200 células ativas de um total de 10.000 células (2% das células estão ativas em qualquer momento). Como podemos lembrar e reconhecer um determinado padrão de 200 células ativas? Uma maneira simples de fazer isso é fazer uma lista das 200 células ativas que nos interessa. Se vemos as mesmas 200 células ativas novamente nós reconhecemos o padrão. No entanto, e se nós fizéssemos uma lista de apenas 20 das 200 células ativas e ignorássemos as outras 180? O que aconteceria? Você pode pensar que se lembrar de apenas 20 células causaria muitos erros pois essas 20 células estariam ativas em muitos padrões diferentes de 200. Mas este não é o caso. Como os padrões são grandes e esparsos (neste exemplo, 200 células ativas de 10.000), lembrar 20 células ativas é quase tão bom como lembrar todas as 200. A chance de erro em um sistema prático é muito pequena e nós temos reduzido nossas necessidades de memória consideravelmente.

As células em uma região HTM aproveitam essa propriedade. Cada um dos segmentos dendríticos de uma célula tem um conjunto de conexões com outras células na região. Um segmento dendrítico forma essas conexões como um meio de reconhecer o estado da rede em algum ponto no tempo. Pode haver centenas ou milhares de células ativas nas proximidades, mas o segmento dendrítico só tem de conectar-se a 15 ou 20 delas. Quando o segmento dendrítico vê 15 destas células ativas, ele tem razoável certeza de que o padrão está ocorrendo. Esta técnica é chamada de "sub-amostragem" e é usada em todos os algoritmos HTM.

Cada célula participa em diversos padrões de distribuição e em muitas seqüências diferentes. Uma célula específica pode ser parte de dezenas ou centenas de transições temporais. Portanto, cada célula tem vários segmentos dendríticos, e não apenas um. O ideal seria se uma célula tivesse um segmento dendrítico para cada padrão de atividade que ela quisesse reconhecer. Na prática, porém, um segmento dendrítico pode aprender conexões para vários padrões completamente diferentes e mesmo assim funcionam bem. Por exemplo, um segmento pode aprender 20 conexões para cada um de quatro padrões diferentes, totalizando 80 conexões. Assim, nós definimos um limite para o segmento dendrítico se tornar ativo quando ao menos 15 de suas conexões estiverem ativas. Isso introduz a possibilidade de erro. É possível, por acaso, que o dendrito atinja seu limite de 15 conexões ativas, misturando partes de diferentes padrões.. No entanto, este tipo de erro é muito improvável, mais uma vez devido à esparsidade das representações.

Agora podemos ver como uma célula com uma ou duas dúzias de segmentos dendríticos e algumas milhares de sinapses pode reconhecer centenas de estados independentes de atividade celular.

Detalhes do agrupador temporal

Vamos enumerar aqui os passos realizados pelo agrupador temporal. Nós começamos de onde o agrupador espacial parou, com um conjunto de colunas ativas que representam a entrada de alimentação.

1) Para cada coluna ativa, checa por células na coluna que estão em um estado preditivo, e ativa-as. Se nenhuma célula está em um estado preditivo, todas as células na coluna são ativadas. O conjunto resultante de células ativas é a representação da entrada no contexto da entrada anterior.

2) Para cada segmento dendrítico em cada célula na região, conta quantas sinapses estabelecidas estão conectadas a células ativas. Se o número exceder um limite, esse segmento dendrítico é marcado como ativo. As células com segmentos dendríticos ativos são colocadas em estado preditivo a menos que elas já estejam ativas devido à entrada de alimentação. Células sem dendritos ativos e não-ativos devido à entrada de baixo-para-cima tornam-se ou permanecem inativas. O conjunto de células agora no estado preditivo é a predição da região.

3) Quando um segmento dendrítico torna-se ativo, modifica os valores de permanência de todas as sinapses associadas ao segmento. Para cada sinapse potencial no segmento dendrítico ativo, aumenta a permanência das sinapses que são conectadas a células ativas e diminui a permanência das sinapses conectadas a células inativas. Essas alterações na permanência da sinapse são marcadas como temporária.

Isso modifica as sinapses em segmentos que já estão treinados o suficiente para tornar o segmento ativo, e assim levar a uma predição. No entanto, nós sempre queremos estender as predições para mais atrás no tempo, se possível. Assim, nós escolhemos um segundo segmento dendrítico na mesma célula para treinar. Para o segundo segmento nós escolhemos o que melhor corresponde ao estado do sistema no intervalo de tempo anterior. Para esse segmento, usando o estado do sistema no intervalo de tempo anterior, é aumentada a permanência das sinapses que estão conectadas a células ativas e diminuída a permanência das sinapses conectadas a células inativas. Essas alterações na permanência da sinapse são marcadas como temporária.

4) Quando uma célula muda de inativa para ativa devido à entrada de alimentação, percorremos cada sinapse potencial associada com a célula e removemos quaisquer marcas temporárias. Assim, nós atualizamos a permanência das sinapses apenas se elas predisseram corretamente a ativação de alimentação da célula.

5) Quando uma célula muda a partir de qualquer estado ativo para inativo, são desfeitas quaisquer alterações de permanência marcadas como temporária para cada sinapse potencial nesta célula. Nós não queremos fortalecer a permanência de sinapses que predizeram incorretamente a ativação de alimentação de uma célula.

Note que apenas as células que estão ativas devido à entrada de alimentação propagam atividade *dentro da* região, do contrário predições levariam a propagação de mais predições. Porém todas as células ativas (de alimentação e preditivas) formam a saída de uma região e a propagam para a *próxima* região na hierarquia.

Predição e seqüências de primeira ordem versus predição e seqüências de ordem variável

Existe mais um tema importante para discutir antes de terminarmos nossa discussão sobre os agrupadores espacial e temporal. Pode não ser de interesse para todos os leitores e não é necessário para entender os capítulos 3 e 4.

Qual é o efeito de ter mais ou menos células por coluna? Especificamente, o que acontece se tivermos uma única célula por coluna?

No exemplo usado anteriormente, nós mostramos que uma representação de uma entrada composta por 100 colunas ativas com 4 células por coluna pode ser codificada em 4^{100} maneiras diferentes. Portanto, a mesma entrada pode aparecer em muitos contextos sem criar confusão. Por exemplo, se os padrões de entrada representam palavras, então uma região pode lembrar muitas frases que usam as mesmas palavras repetidas várias vezes e que não se confundem. Uma palavra como "cachorro" poderia ter uma representação única em diferentes contextos. Esta capacidade permite que uma região HTM faça o que é chamado de predições de "ordem variável".

A predição de ordem variável não se baseia unicamente no que está acontecendo atualmente, mas em quantidades variáveis do contexto passado. Uma região HTM é uma memória de ordem variável.

Se aumentarmos para cinco células por coluna, o número disponível de codificações de qualquer entrada no nosso exemplo poderia aumentar para 5^{100} , um aumento enorme sobre 4^{100} . Mas ambos os números são tão grandes que para muitos problemas práticos o aumento da capacidade não pode ser útil.

No entanto, fazer o número de células por coluna muito menor faz uma grande diferença.

Se nós percorrermos todo o caminho em uma célula por coluna, perderemos a capacidade de contemplar o contexto em nossas representações. Uma entrada para uma região sempre resultará na mesma predição, independentemente da atividade anterior. Com uma célula por coluna, a memória de uma região HTM será uma memória de "primeira ordem"; predições serão baseadas apenas na entrada corrente.

A predição de primeira ordem é ideal para um tipo de problema que os cérebros resolvem: inferência espacial estática. Como afirmado anteriormente, um ser humano exposto a uma breve imagem visual pode reconhecer qual é o objeto mesmo que a exposição seja muito curta a ponto dos olhos se moverem. Com a audição, você sempre precisará ouvir uma seqüência de padrões para reconhecer qual é. A visão é geralmente assim, você costuma processar um fluxo de imagens visuais. Mas, sob certas condições, você pode reconhecer uma imagem com uma única exposição.

Reconhecimento temporal e estático podem parecer exigir mecanismos de inferência diferentes. Um requer o reconhecimento de seqüências de padrões e faz predições com base no contexto de extensão variável. O outro exige o reconhecimento de um padrão espacial estático sem usar o contexto temporal. Uma região HTM com várias células por coluna é ideal para o reconhecimento de seqüências baseadas no tempo, e uma região HTM com uma célula por coluna é ideal para reconhecimento de padrões espaciais. Na Numenta, temos realizado muitos experimentos usando regiões de uma-célula-por-coluna aplicados à problemas de visão. Os detalhes desses experimentos estão além do escopo deste capítulo; mas nós cobriremos os conceitos importantes.

Se expomos uma região HTM a imagens, as colunas na região aprendem a representar arranjos espaciais comuns dos pixels. O tipo de padrões aprendidos são semelhantes ao que é observado pela região V1 do neocórtex (uma região neocortical extensivamente estudada na biologia), normalmente linhas e curvas em diferentes orientações. Se for treinada em imagens se movimentando, a região HTM aprende transições dessas formas básicas. Por exemplo, uma linha vertical em uma posição é geralmente seguida por uma linha vertical deslocada para a esquerda ou para a direita. Todas as transições dos padrões comumente observados são lembrados pela região HTM.

Agora o que acontece se nós expormos a região a uma imagem de uma linha vertical que se desloca para a direita? Se a nossa região tem apenas uma célula por coluna, ela vai prever que a próxima linha poderá aparecer à esquerda ou à direita. Ela não pode usar o contexto para saber onde a linha estava no passado e portanto saber se ela está se movendo para a esquerda ou para a direita. O que você encontra é que essas células de uma-célula-por-coluna comportam-se como "células complexas" no neocórtex. A saída preditiva de tal célula ficará ativa por uma linha visível em diferentes posições, independentemente se a linha está se movendo para a esquerda ou para a direita ou para nenhum dos dois lados. Temos ainda observado que uma região como esta exibe estabilidade na tradução, mudanças de escala, etc, mantendo a capacidade de distinguir entre diferentes imagens. Esse comportamento é o que é necessário para a invariância espacial (que reconhece o mesmo padrão em diferentes locais de uma imagem).

Se nós agora fizermos a mesma experiência em uma região HTM com várias células por coluna, descobriremos que as células se comportam como "células complexas direcionalmente-afinadas" no neocórtex. A saída preditiva de uma célula ficará ativa por uma linha que se desloca para a esquerda ou uma linha que se desloca para a direita, mas não ambas.

Colocando tudo isso junto, faremos a seguinte hipótese. O neocórtex faz tanto a predição e inferência de primeira ordem como a de ordem variável. Há quatro ou cinco camadas de células em cada região do neocórtex. As camadas diferem de várias maneiras mas todas elas têm propriedades de resposta colunar compartilhadas e grande conectividade horizontal dentro da camada. Nós especulamos que cada camada de células no neocórtex está realizando uma variação da inferência e das regras de aprendizagem HTM descritas neste capítulo. As diferentes camadas de células desempenham diferentes papéis. Por exemplo, é conhecido a partir de

estudos anatômicos que a camada 6 cria realimentação na hierarquia e que a camada 5 está envolvida no comportamento motor. As duas principais camadas de alimentação de células são as camadas 4 e 3. Nós especulamos que uma das diferenças entre as camadas 4 e 3 é que as células da camada 4 estão atuando de forma independente, ou seja, uma célula por coluna, enquanto que as células da camada 3 estão atuando como várias células por coluna. Assim, as regiões do neocórtex próximas à entrada sensorial têm tanto memória de primeira ordem como de ordem variável. A memória de sequência de primeira ordem (que corresponde aproximadamente aos neurônios da camada 4) é útil na formação de representações que são invariantes diante das transformações espaciais. A memória de sequência de ordem variável (que corresponde aproximadamente aos neurônios da camada 3) é útil para a inferência e predição de imagens em movimento.

Em resumo, nós usamos a hipótese de que os algoritmos similares aos descritos neste capítulo estão trabalhando em todas as camadas de neurônios no neocórtex. As camadas no neocórtex variam em detalhes importantes que fazem ele desempenhar diferentes papéis relacionados com a alimentação versus realimentação, atenção, e comportamento motor. Em regiões próximas a estímulos sensoriais, é útil ter uma camada de neurônios realizando memória de primeira ordem de forma a isso levar a invariância espacial.

Na Numenta, temos experimentado regiões HTM de primeira ordem (uma única célula por coluna) para problemas de reconhecimento de imagem. Nós também temos experimentado regiões HTM de ordem variável (várias células por coluna) para reconhecer e prever sequências de ordem variável. No futuro, seria lógico tentar combiná-las em uma única região e ampliar os algoritmos para outros fins. No entanto, acreditamos que muitos problemas interessantes podem ser tratados com o equivalente das regiões de uma-única-célula-por-coluna e das regiões de múltiplas-células-por-coluna, estando sozinhas ou em uma hierarquia.

Capítulo 3: Pseudocódigo e Implementação do Agrupamento Espacial

Este capítulo contém o pseudocódigo detalhado para uma primeira implementação da função agrupador espacial. A entrada para este código é uma matriz de entradas binárias de baixo-para-cima a partir de dados sensoriais ou do nível anterior. O código calcula `activeColumns(t)` - a lista de colunas que vencem devido à entrada de baixo-para-cima no tempo t . Essa lista é enviada como entrada para a rotina agrupador temporal descrita no próximo capítulo, ou seja, `activeColumns(t)` é a saída da rotina de agrupamento espacial.

O pseudocódigo é dividido em três fases distintas que ocorrem em seqüência:

- Fase 1: calcula a sobreposição com a entrada atual para cada coluna
- Fase 2: calcula as colunas vencedoras após a inibição
- Fase 3: atualiza a permanência da sinapse e as variáveis internas

Embora a aprendizagem do agrupador espacial seja inerentemente on-line, você poderá desativar a aprendizagem por simplesmente pular a Fase 3.

O restante do capítulo contém pseudocódigo para cada uma das três etapas. As variadas estruturas de dados e rotinas de suporte usadas no código são definidas no final.

Inicialização

Antes de receber quaisquer entradas, a região é inicializada por se computar uma lista inicial das sinapses potenciais de cada coluna. Isto consiste de um conjunto aleatório de entradas selecionadas a partir do espaço de entrada. Cada entrada é representada por uma sinapse e lhe é atribuído um valor de permanência aleatório. Os valores de permanência aleatórios são escolhidos com base em dois critérios. Primeiro, os valores são escolhidos para estar em uma pequena escala em torno de `connectedPerm` (o valor mínimo de permanência em que uma sinapse é considerada "conectada"). Isso permite que sinapses potenciais se tornem conectadas (ou desconectadas) depois de um pequeno número de iterações de treinamento. Segundo, cada coluna tem um centro natural sobre a região de entrada, e os valores de permanência possuem uma tendência a esse centro (eles possuem valores mais elevados perto do centro).

Fase 1: Sobreposição

Dado um vetor de entrada, a primeira fase calcula a sobreposição de cada coluna com o vetor. A sobreposição de cada coluna é simplesmente o número de sinapses relacionadas com as entradas ativas, multiplicado pelo respectivo estímulo. Se este valor for inferior a `minOverlap`, definimos o valor de sobreposição para zero.

```

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)

```

Fase 2: Inibição

A segunda fase calcula as colunas que permanecem como vencedoras após a etapa de inibição. `desiredLocalActivity` é um parâmetro que controla o número de colunas que terminaram vencendo. Por exemplo, se `desiredLocalActivity` é 10, uma coluna será vencedora se o valor de sobreposição for maior do que o valor das dez colunas com valores mais altos dentro de seu raio de inibição.

```

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.

```

Fase 3: Aprendizagem

A terceira fase realiza a aprendizagem; ela atualiza os valores de permanência de todas as sinapses se necessário, bem como o estímulo e o raio de inibição.

A principal regra de aprendizagem é implementada nas linhas 20-26. Para colunas vencedoras, se uma sinapse estiver ativa, seu valor de permanência é aumentado, caso contrário, é diminuído. Valores de permanência são obrigados a estar entre 0 e 1.

As linhas 28-36 implementam o estímulo. Existem dois mecanismos de estímulo distintos que ajudam uma coluna a aprender conexões. Se uma coluna não vence o suficiente (conforme medido por `activeDutyCycle`), seu valor de estímulo global é aumentado (linha 30-32). Alternativamente, se as sinapses conectadas de uma coluna não se sobrepõem bem com quaisquer entradas em várias vezes suficientes (conforme medido por `overlapDutyCycle`), os seus valores de permanência são estimulados (linha 34-36). Nota: uma vez que a aprendizagem é desligada, `boost(c)` é congelado.

Finalmente, no final da Fase 3, o raio de inibição é recalculado (linha 38).

```

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.

```

Estruturas de dados e rotinas de suporte

As seguintes variáveis e estruturas de dados são usadas no pseudocódigo:

columns	Lista de todas as colunas.
input(t, j)	A entrada para este nível no tempo t. input(t, j) é 1 se a j'ésima entrada estiver ligada.
overlap(c)	O agrupador espacial sobrepõe a coluna c com um padrão de entrada específico.
activeColumns(t)	Lista dos índices das colunas que são vencedoras devido à entrada de baixo-para-cima.
desiredLocalActivity	Um parâmetro que controla o número de colunas que serão vencedoras após a etapa de inibição.
inhibitionRadius	Média do tamanho do campo receptivo conectado das colunas.
neighbors(c)	Uma lista de todas as colunas que estão dentro do inhibitionRadius da coluna c.
minOverlap	Um número mínimo de entradas que devem estar ativas para que uma coluna seja considerada durante a fase de inibição.
boost(c)	O valor de estímulo para a coluna c calculado durante a aprendizagem - utilizado para aumentar o valor de sobreposição das colunas inativas.

<code>synapse</code>	Uma estrutura de dados que representa uma sinapse - contém um valor de permanência e o índice da entrada de origem.
<code>connectedPerm</code>	Se o valor de permanência de uma sinapse for superior a este valor, será dito que ela está conectada.
<code>potentialSynapses(c)</code>	A lista de sinapses potenciais e seus valores de permanência.
<code>connectedSynapses(c)</code>	Um subconjunto de <code>potentialSynapses(c)</code> quando o valor de permanência for superior a <code>connectedPerm</code> . Estas são as entradas de baixo-para-cima que atualmente estão conectadas à coluna <code>c</code> .
<code>permanenceInc</code>	Quantidade de valores de permanência das sinapses que é aumentado durante a aprendizagem.
<code>permanenceDec</code>	Quantidade de valores de permanência das sinapses que é diminuído durante a aprendizagem.
<code>activeDutyCycle(c)</code>	Uma média variável que representa quantas vezes a coluna <code>c</code> tem estado ativa após a inibição (por exemplo, ao longo das últimas 1000 iterações).
<code>overlapDutyCycle(c)</code>	Uma média variável que representa quantas vezes a coluna <code>c</code> teve sobreposição significativa (ou seja, maior do que <code>minOverlap</code>) com suas entradas (por exemplo, ao longo das últimas 1000 iterações).
<code>minDutyCycle(c)</code>	Uma variável que representa a taxa mínima de disparo desejada para uma célula. Se a taxa de disparo de uma célula for inferior a esse valor, ela será aumentada. Este valor é calculado como 1% da taxa máxima de disparo de suas vizinhas.

As seguintes rotinas de suporte são usadas no código acima.

`kthScore(cols, k)`

Dada a lista de colunas, retorna o k'ésimo maior valor de sobreposição.

`updateActiveDutyCycle(c)`

Calcula uma média móvel de quantas vezes a coluna `c` tem estado ativa após a inibição.

`updateOverlapDutyCycle(c)`

Calcula uma média móvel de quantas vezes a coluna `c` tem sobreposição maior do que `minOverlap`.

`averageReceptiveFieldSize()`

O raio da média do tamanho do campo receptivo conectado de todas as colunas. O tamanho do campo receptivo conectado de uma coluna inclui apenas as sinapses conectadas (aquelas com valores de permanência \geq `connectedPerm`). Isto é usado para determinar a extensão da inibição lateral entre as colunas.

`maxDutyCycle(cols)`

Retorna o ciclo ativo máximo das colunas na lista de colunas dada.

`increasePermanences(c, s)`

Aumenta o valor de permanência de cada sinapse na coluna `c` por um fator de escala `s`.

`boostFunction(c)`

Retorna o valor de estímulo de uma coluna. O valor de estímulo é um número escalar ≥ 1 . Se `activeDutyCycle(c)` está acima de `minDutyCycle(c)`, o valor de estímulo é 1. O estímulo aumenta linearmente a cada vez que o `activeDutyCycle` de uma coluna começa a cair para abaixo do seu `minDutyCycle`.

Capítulo 4: Pseudocódigo e Implementação do Agrupamento Temporal

Este capítulo contém o pseudocódigo detalhado para uma primeira implementação da função agrupador temporal. A entrada para este código é `activeColumns(t)`, computada pelo agrupador espacial. O código computa o estado ativo e preditivo para cada célula na iteração atual, t . O OR booleano dos estados ativo e preditivo para cada célula constitui a saída do agrupador temporal para o próximo nível.

O pseudocódigo é dividido em três fases distintas que ocorrem em seqüência:

Fase 1: calcula o estado ativo, `activeState(t)`, para cada célula

Fase 2: calcula o estado preditivo, `predictiveState(t)`, para cada célula

Fase 3: atualiza as sinapses

A Fase 3 só é necessária para a aprendizagem. No entanto, ao contrário do agrupamento espacial, as Fases 1 e 2 contêm algumas operações de aprendizagem-específica quando a aprendizagem é ligada. Partindo do pressuposto que agrupamento temporal é significativamente mais complicado do que agrupamento espacial, iremos listar primeiro somente a versão de inferência do agrupador temporal, seguida por uma versão que combina inferência e aprendizagem. A descrição de alguns dos detalhes de implementação, terminologia, e rotinas de suporte está no final do capítulo, após o pseudocódigo.

Pseudocódigo do agrupador temporal: inferência isolada

Fase 1

A primeira fase calcula o estado ativo para cada célula. Para cada coluna vencedora vamos determinar quais células devem tornar-se ativas. Se a entrada de baixo-para-cima foi predita por qualquer célula (ou seja, seu `predictiveState` foi 1 devido a um segmento de seqüência no intervalo de tempo anterior), então essas células se tornam ativas (linhas 4-9). Se a entrada de baixo-para-cima foi inesperada (ou seja, nenhuma célula teve `predictiveState` como ligado), então cada célula na coluna se torna ativa (linhas 11-13).

```

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

```

Fase 2

A segunda fase calcula o estado preditivo para cada célula. Uma célula deixará seu `predictiveState` ligado se qualquer um dos seus segmentos se tornar ativo, ou seja, se um número suficiente de suas conexões horizontais estiver disparando devido à entrada de alimentação.

```

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

```

Pseudocódigo do agrupador temporal: inferência e aprendizagem combinadas

Fase 1

A primeira fase calcula o `activeState` para cada célula que está em uma coluna vencedora. Para essas colunas, o código avança selecionando uma célula de cada coluna para ser a célula de aprendizagem (`learnState`). A lógica é a seguinte: se a entrada de baixo-para-cima foi predita por qualquer célula (ou seja, se a saída de `predictiveState` estava como 1 devido a um segmento de sequência), então essas células se tornam ativas (linhas 23-27). Se esse segmento tornou-se ativo a partir de células escolhidas com `learnState`, esta célula é selecionada como a célula de aprendizagem (linhas 28-30). Se a entrada de baixo-para-cima não estava predita, então todas as células tornam-se ativas (linhas 32-34). Além disso, a melhor célula correspondente é escolhida como a célula de aprendizagem (linhas 36-41) e um novo segmento é adicionado à célula.


```

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.         i,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)

```

Fase 2

A segunda fase calcula o estado preditivo para cada célula. Uma célula deixará ligada a sua saída do estado preditivo se um dos seus segmentos se tornar ativo, ou seja, se um número suficiente das suas entradas laterais estiverem atualmente ativas devido à entrada de alimentação. Neste caso, as células fazem as seguintes mudanças: a) reforço do segmento atualmente ativo (linhas 47-48), e b) reforço de um segmento que poderia ter predito esta ativação, ou seja, um segmento que tem uma correspondência (potencialmente fraca) à atividade ocorrida durante o intervalo de tempo anterior (linhas 50-53).

```

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)

```

Fase 3

É na terceira e última fase, que a aprendizagem é realmente realizada. Nesta fase as atualizações do segmento que tinham sido enfileiradas são realmente implementadas assim que nós obtemos

a entrada de alimentação e a célula é escolhida como uma célula de aprendizagem (linhas 56-57). Caso contrário, se a célula para de prever por qualquer motivo, nós negativamente reforçamos os segmentos (linhas 58-60).

```
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.
```

Detalhes e terminologia de implementação

Nesta seção, descrevemos alguns dos detalhes da implementação do nosso agrupador temporal e a terminologia. Cada célula é indexada utilizando dois números: um índice da coluna, *c*, e um índice da célula, *i*. As células mantêm uma lista de segmentos dendríticos, onde cada segmento contém uma lista de sinapses mais um valor de permanência de cada sinapse. Mudanças nas sinapses de uma célula são marcadas como temporárias até a célula tornar-se ativa a partir da entrada de alimentação. Estas alterações temporárias são mantidas em `segmentUpdateList`. Cada segmento tem também um sinalizador booleano, `sequenceSegment`, que indica se o segmento predisser a entrada de alimentação para o próximo intervalo de tempo.

A implementação de sinapses potenciais é diferente da implementação no agrupador espacial. No agrupador espacial, a lista completa das sinapses potenciais é representada como uma lista explícita. No agrupador temporal, cada segmento pode ter sua própria lista (possivelmente grande) de sinapses potenciais. Na prática a manutenção de uma longa lista para cada segmento é computacionalmente cara e consome muita memória. Portanto, no agrupador temporal, nós adicionamos aleatoriamente sinapses ativas para cada segmento durante a aprendizagem (controlada pelo parâmetro `newSynapseCount`). Essa otimização tem um efeito similar a manter uma lista completa de sinapses potenciais, porém a lista por segmento é muito menor mantendo a possibilidade de aprender novos padrões temporais.

O pseudocódigo também usa uma pequena máquina de estado para acompanhar os estados das células em intervalos de tempo diferentes. Nós mantemos três estados diferentes para cada célula. As matrizes `activeState` e `predictiveState` acompanham os estados ativo e preditivo de cada célula em cada intervalo de tempo. A matriz `learnState` determina quais saídas celulares serão usadas durante a aprendizagem. Quando uma entrada é inesperada, todas as células em uma coluna específica tornam-se ativas no mesmo intervalo de tempo. Apenas uma destas células (a célula que melhor combina com a entrada) tem o seu `learnState` ligado. Nós só adicionamos as sinapses das células que têm seu `learnState` definido para 1 (isso evita que se represente mais de uma coluna totalmente ativa em segmentos dendríticas).

As estruturas de dados a seguir são usadas no pseudocódigo do agrupador temporal:

`cell(c, i)` Uma lista de todas as células, indexadas por *i* e *c*.
`cellsPerColumn` Número de células em cada coluna.

<code>activeColumns(t)</code>	Lista de índices de colunas que são vencedoras devido à entrada de baixo-para-cima (isto é, a saída do agrupador espacial).
<code>activeState(c, i, t)</code>	Um vetor booleano com um número por célula. Ele representa o estado ativo da célula <i>i</i> da coluna <i>c</i> no tempo <i>t</i> , dada a entrada de alimentação atual e o contexto temporal do passado. <code>activeState(c, i, t)</code> é a contribuição da célula <i>i</i> da coluna <i>c</i> no tempo <i>t</i> . Se for 1, é por que a célula possui entrada de alimentação atual, bem como um contexto temporal apropriado.
<code>predictiveState(c, i, t)</code>	Um vetor booleano com um número por célula. Ele representa a predição da célula <i>i</i> da coluna <i>c</i> no tempo <i>t</i> , dada a atividade de baixo-para-cima das outras colunas e o contexto temporal do passado. <code>predictiveState(c, i, t)</code> é a contribuição da célula <i>c</i> da coluna <i>i</i> no tempo <i>t</i> . Se for 1, é por que a célula está predizendo entrada de alimentação no contexto temporal atual.
<code>learnState(c, i, t)</code>	Um booleano que indica se a célula <i>i</i> da coluna <i>c</i> é escolhida como a célula da aprendizagem.
<code>activationThreshold</code>	Limite de ativação para um segmento. Se o número de sinapses ativas conectadas em um segmento for maior do que <code>activationThreshold</code> , o segmento será considerado ativo.
<code>learningRadius</code>	A área em torno de uma célula do agrupador temporal a partir da qual pode se obter conexões laterais.
<code>initialPerm</code>	Valor de permanência inicial para uma sinapse.
<code>connectedPerm</code>	Se o valor de permanência de uma sinapse for superior a este valor, será dito que ela está conectada.
<code>minThreshold</code>	Atividade de segmento mínima para a aprendizagem.
<code>newSynapseCount</code>	O número máximo de sinapses adicionadas a um segmento durante a aprendizagem.
<code>permanenceInc</code>	Quantidade de valores de permanência das sinapses que é aumentado durante a aprendizagem.
<code>permanenceDec</code>	Quantidade de valores de permanência das sinapses que é diminuído durante a aprendizagem.
<code>segmentUpdate</code>	Estrutura de dados que armazena três peças de informação necessária para atualizar um determinado segmento: a) o índice do segmento (-1 se ele for um novo segmento), b) uma lista de sinapses ativas existentes, e c) um sinalizador que indica se este segmento deve ser marcado como um segmento de sequência (o padrão é false).
<code>segmentUpdateList</code>	Uma lista de estruturas de <code>segmentUpdate</code> . <code>segmentUpdateList(c, i)</code> é a lista de mudanças para a célula <i>i</i> da coluna <i>c</i> .

As seguintes rotinas de suporte são usadas no código acima:

`segmentActive(s, t, state)`

Esta rotina retorna **true** se o número de sinapses conectadas no segmento `s` que está ativo devido à determinado estado no tempo `t` é maior do que `activationThreshold`. O parâmetro `state` pode ser `activeState`, ou `learnState`.

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

Dada uma célula `c` da coluna `i`, retorna um índice de segmento de tal forma que `segmentActive(s, t, state)` será **true**. Se vários segmentos estiverem ativos, será dada a preferência aos segmentos de seqüência. Caso contrário, será dada a preferência aos segmentos com maior atividade.

`getBestMatchingSegment(c, i, t)`

Dada uma célula `c` da coluna `i` no tempo `t`, encontra o segmento com o maior número de sinapses ativas. Essa rotina é agressiva na busca do melhor resultado. É permitido que o valor de permanência das sinapses esteja abaixo de `connectedPerm`. É permitido que o número de sinapses ativas seja inferior a `activationThreshold`, porém ele deverá estar acima de `minThreshold`. A rotina retorna o índice do segmento. Se nenhum segmento for encontrado, então um índice de -1 será retornado.

`getBestMatchingCell(c)`

Dada uma coluna de dados, retorna a célula com o melhor segmento de correspondência (conforme definido acima). Se nenhuma célula tiver um segmento correspondente, será retornado a célula com o menor número de segmentos.

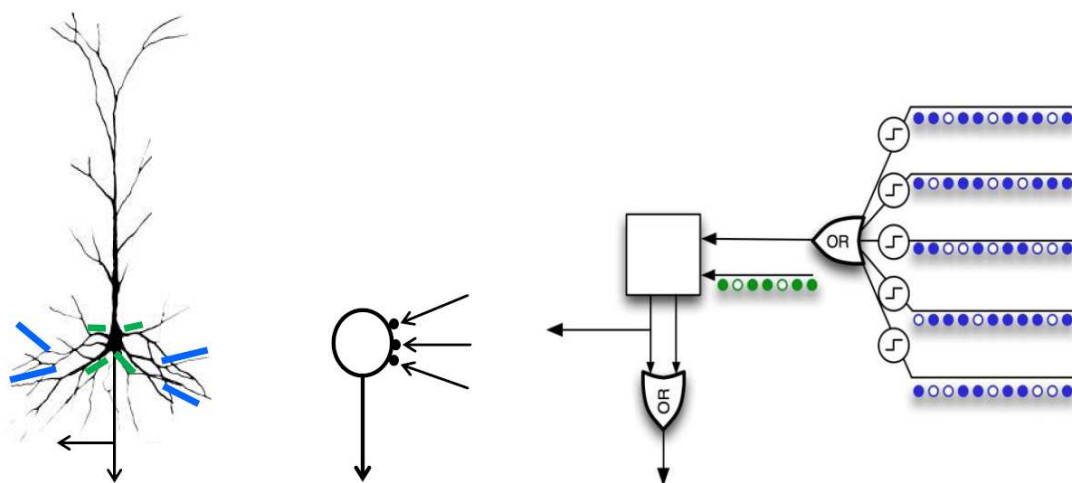
`getSegmentActiveSynapses(c, i, t, s, newSynapses = false)`

Retorna uma estrutura de dados de `segmentUpdate` contendo uma lista de mudanças propostas para o segmento `s`. Deixa `activeSynapses` ser a lista de sinapses ativas quando as células de origem terem a saída `activeState = 1` no intervalo de tempo `t`. (Esta lista será vazia se `s = -1`, uma vez que o segmento não existe.) `NewSynapses` é um argumento opcional onde o padrão é **false**. Se `newSynapses` for **true**, então o resultado da subtração `newSynapseCount - count(activeSynapses)` será adicionado à `activeSynapses`. Essas sinapses são escolhidas aleatoriamente a partir do conjunto de células que têm a saída `learnState = 1` no intervalo de tempo `t`.

`adaptSegments(segmentList, positiveReinforcement)`

Esta função itera através de uma lista de `segmentUpdate` e reforça cada segmento. Para cada elemento `segmentUpdate`, as seguintes alterações são realizadas. Se `positiveReinforcement` for **true**, então as sinapses na lista ativa tem a sua contagem de permanência aumentada por `permanenceInc`. Todas as outras sinapses tem a sua contagem de permanência diminuída por `permanenceDec`. Se `positiveReinforcement` for **false**, então as sinapses na lista ativa tem a sua contagem de permanência diminuída por `permanenceDec`. Após este passo, quaisquer sinapses no `segmentUpdate` que ainda não existem são adicionadas com uma contagem de permanência no valor de `initialPerm`.

Apêndice A: Uma Comparação entre Neurônios Biológicos e Células HTM



A imagem acima mostra uma foto de um neurônio biológico no lado esquerdo, um neurônio artificial simples no meio, e um neurônio HTM ou "célula" do lado direito. O objetivo deste apêndice é fornecer uma melhor compreensão das células HTM e como elas funcionam, comparando-as aos neurônios reais e aos neurônios artificiais mais simples.

Neurônios reais são tremendamente complicados e variados. Vamos concentrar-nos nos princípios mais gerais e somente naqueles que se aplicam ao nosso modelo. Apesar de ignorarmos muitos detalhes dos neurônios reais, as células usadas nos algoritmos de aprendizagem cortical HTM são muito mais realísticas do que os neurônios artificiais utilizados na maioria das redes neurais. Todos os elementos incluídos nas células HTM são necessários para o funcionamento de uma região HTM.

Neurônios biológicos

Neurônios são células que transportam informações no cérebro. A imagem acima na esquerda é de um típico neurônio excitatório. A aparência visual de um neurônio é dominada pelo dendritos de ramificação. Todas as entradas para um neurônio excitatório através de sinapses são alinhadas ao longo dos dendritos. Nos anos recentes nosso conhecimento dos neurônios avançou consideravelmente. A maior mudança foi em perceber que os dendritos de um neurônio não são apenas condutores para levar entradas para o corpo celular. Nós sabemos agora que os dendritos são complexos elementos de processamento não-linear em si. Os algoritmos de aprendizagem cortical HTM tiram proveito dessas propriedades não-lineares. Os neurônios têm várias partes.

Corpo celular

O corpo da célula é o pequeno volume no centro do neurônio. A saída da célula, o axônio, origina-se no corpo celular. As entradas para a célula são as sinapses alinhadas ao longo dos dendritos os quais alimentam o corpo celular.

Dendritos Proximais

Os ramos de dendrito mais próximos do corpo da célula são chamados de dendritos proximais. No diagrama alguns dendritos proximais estão marcados com linhas verdes.

Várias sinapses ativas em dendritos proximais possuem um efeito aditivo quase linear no corpo celular. Cinco sinapses ativas conduzirão a despolarização no corpo celular aproximadamente cinco vezes mais em comparação com uma sinapse ativa. Em contraste, se uma única sinapse é ativada repetidamente por uma sucessão rápida de potenciais de ação, o segundo, o terceiro, e os subsequentes potenciais de ação terão muito menos efeito no corpo celular, que o primeiro.

Portanto, podemos dizer que as entradas para os dendritos proximais são somadas linearmente no corpo da célula, e picos rápidos que chegam em uma única sinapse terão apenas um efeito ligeiramente maior do que um único pico.

As conexões de alimentação para uma região do neocórtex se conectam preferencialmente em dendritos proximais. Isto foi relatado pelo menos para os neurônios da camada 4, a principal camada de neurônios de entrada em cada região.

Dendritos Distais

Os ramos de dendritos mais longe do corpo celular são chamados de dendritos distais. No diagrama alguns dendritos distais estão marcados com linhas azuis.

Dendritos distais são mais finos que os dendritos proximais. Eles se conectam a outros ramos de dendrito na árvore dendrítica e não se conectam diretamente ao corpo celular. Estas diferenças dão aos dendritos distais propriedades elétricas e químicas únicas. Quando uma única sinapse é ativada em um dendrito distal, ela possui um efeito mínimo no corpo celular. A despolarização que ocorre localmente para a sinapse enfraquece no momento em que ela atinge o corpo celular. Por muitos anos isto foi visto como um mistério. Parecia que as sinapses distais, que são a maioria das sinapses em um neurônio, não podiam fazer muita coisa.

Sabemos agora que as seções de dendritos distais atuam como regiões de processamento semi-independentes. Se um número suficiente de sinapses se tornar ativo ao mesmo tempo dentro de uma curta distância ao longo do dendrito, elas poderão gerar um pico dendrítico que poderá viajar para o corpo da célula com grande efeito. Por exemplo, vinte sinapses ativas dentro de 40 microns uma da outra irão gerar um pico dendrítico.

Portanto, podemos dizer que os dendritos distais atuam como um conjunto de detectores de limites de coincidência.

As sinapses formadas em dendritos distais são predominantemente de outras células vizinhas na região.

A imagem mostra um grande ramo de dendrito estendendo para cima que é chamado de dendrito apical. Uma teoria diz que essa estrutura permite ao neurônio localizar vários dendritos distais em uma área onde eles possam fazer facilmente mais conexões com os axônios passageiros. Nessa interpretação, o dendrito apical atua como uma extensão da célula.

Sinapses

Um neurônio típico pode ter várias milhares de sinapses. A grande maioria (talvez 90%) destas será em dendritos distais, e o restante em dendritos proximais.

Por muitos anos pensava-se que a aprendizagem envolvia fortalecimento e enfraquecimento do efeito ou "peso" das sinapses. Embora este efeito tenha sido observado, cada sinapse é um pouco estocástica. Quando ativada, ela liberará de forma não confiável um neurotransmissor. Portanto, os algoritmos utilizados pelo cérebro não podem depender da precisão ou da fidelidade dos pesos de sinapses individuais.

Além disso, sabemos agora que sinapses inteiras se formam e se desformam rapidamente. Esta flexibilidade representa uma forma poderosa de aprendizagem e explica melhor a aquisição rápida de conhecimentos. Uma sinapse só pode se formar se um axônio e um dendrito estiverem dentro de uma certa distância, conduzindo ao conceito de sinapses "potenciais". Com estes pressupostos, a aprendizagem ocorre principalmente através da formação de sinapses válidas a partir de sinapses potenciais.

Saída do Neurônio

A saída de um neurônio é um pico, ou "potencial de ação", que se propaga ao longo do axônio. O axônio deixa o corpo celular e quase sempre se divide em dois. Um ramo viaja horizontalmente fazendo muitas conexões com outras células vizinhas. O outro ramo se projeta para outras camadas de células ou para outras partes do cérebro. Na imagem do neurônio acima, o axônio não está visível. Nós adicionamos uma linha e duas setas para representar esse axônio.

Apesar da saída efetiva de um neurônio ser sempre um pico, existem diferentes visões sobre como interpretar isso. A visão predominante (especialmente no que diz respeito ao neocórtex) é que a taxa de picos é o que importa. Portanto, a saída de uma célula pode ser vista como um valor escalar.

Alguns neurônios também apresentam um comportamento de "estouro", uma série curta e rápida de uns poucos picos que são diferentes do padrão de picagem regular.

A descrição acima de um neurônio se destina a dar uma breve introdução aos neurônios. Ela se concentra nos atributos que correspondem às características das células HTM e deixa de fora muitos detalhes. Nem todas as características que acabamos de descrever são universalmente aceitas. Nós as incluímos porque são necessárias para os nossos modelos. O que se sabe sobre os neurônios poderia facilmente encher vários livros, e pesquisas ativas sobre eles continuam até hoje.

Neurônios artificiais simples

A imagem do meio no início deste apêndice apresenta um elemento similar a um neurônio utilizado em muitos modelos de redes neurais artificiais clássicas. Esses neurônios artificiais possuem um conjunto de sinapses, cada uma com um peso. Cada sinapse recebe uma ativação escalar, que é multiplicada pelo peso da sinapse. As saídas de todas as sinapses são somadas de uma forma não-linear para produzir a saída do neurônio artificial. A aprendizagem ocorre através do ajuste dos pesos das sinapses e talvez por função não-linear.

Esse tipo de neurônio artificial, e as suas variações, tem se mostrado útil em muitas aplicações como uma valiosa ferramenta computacional. No entanto, ele não captura grande parte da complexidade e poder de processamento dos neurônios biológicos. Se quisermos compreender e modelar como um conjunto de neurônios reais funciona no cérebro precisamos de um modelo de neurônio mais sofisticado.

Células HTM

Em nossa ilustração, a imagem da direita mostra uma célula usada nos algoritmos de aprendizagem cortical HTM. Uma célula HTM captura muitas das capacidades importantes dos neurônios reais, mas também faz várias simplificações.

Dendrito Proximal

Cada célula HTM tem um único dendrito proximal. Todas as entradas de alimentação para a célula são feitas através de sinapses (mostradas como pontos verdes). A atividade das sinapses é linearmente somada para produzir uma ativação de alimentação para a célula.

Exigimos que todas as células em uma coluna tenham a mesma resposta de alimentação. Em neurônios reais, isto provavelmente seria feito por um tipo de célula inibitória. Em HTMs nós simplesmente forçamos todas as células em uma coluna a compartilhar um único dendrito proximal.

Para evitar ter células que nunca vençam a competição com as células vizinhas, uma célula HTM estimulará a sua ativação de alimentação se não estiver vencendo o suficiente em relação as suas vizinhas. Assim, há uma constante competição entre as células. Novamente, em uma HTM nós modelamos isto como uma competição entre colunas, não células. Esta competição não está ilustrada no diagrama.

Finalmente, o dendrito proximal tem um conjunto associado de sinapses potenciais que é um subconjunto de todas as entradas para uma região. A medida que a célula aprende, ela aumenta ou diminui o valor de "permanência" de todas as sinapses potenciais no dendrito proximal. Somente as sinapses potenciais que estão acima de um limite são válidas.

Conforme mencionado anteriormente, o conceito de sinapses potenciais vem da biologia para se referir à axônios e dendritos que estão próximos o suficiente para formar uma sinapse. Nós estendemos esse conceito para um conjunto maior de conexões possíveis para uma célula HTM. Dendritos e axônios dos neurônios biológicos podem crescer e se retratar a medida que a aprendizagem ocorre, e portanto, o conjunto de sinapses potenciais mudam com o crescimento. Ao fazer o conjunto de sinapses potenciais em uma célula HTM grande, nós atingimos aproximadamente o mesmo resultado a medida que os axônios e dendritos crescem. O conjunto de sinapses potenciais não é mostrado.

A combinação da concorrência entre as colunas, da aprendizagem a partir de um conjunto de sinapses potenciais, e da estimulação de colunas subutilizadas dá a uma região de neurônios HTM uma poderosa plasticidade que também é vista no cérebro. Uma região HTM irá ajustar automaticamente o que cada coluna representará (por meio de mudanças nas sinapses nos dendritos proximais) caso a entrada mude, ou caso o número de colunas aumente ou diminua.

Dendritos Distais

Cada célula HTM mantém uma lista de segmentos dendríticos distais. Cada segmento atua como um detector de limite. Se o número de sinapses ativas em qualquer segmento (mostrados como pontos azuis no diagrama anterior) está acima de um limite, o segmento se torna ativo, e as células associadas entram em estado preditivo. O estado preditivo de uma célula é o OR das ativações dos seus segmentos.

Um segmento dendrítico lembra o estado da região por formar conexões com células que estavam ativas juntas em um ponto no tempo. O segmento se lembra de um estado que

antecedia a célula tornando-a ativa devido a entrada de alimentação. Assim, o segmento está procurando por um estado que prediga que sua célula se tornará ativa. Um limite típico para um segmento dendrítico é 15. Se 15 sinapses válidas em um segmento estiverem ativas ao mesmo tempo, o dendrito se torna ativo. Pode haver centenas ou milhares de células ativas nas proximidades, mas a conexão de apenas 15 é suficiente para reconhecer o padrão mais amplo.

Cada segmento dendrítico distal também tem conjunto associado de sinapses potenciais. O conjunto de sinapses potenciais é um subconjunto de todas as células em uma região. A medida que o segmento aprende, ele aumenta ou diminui o valor de permanência de todas as suas sinapses potenciais. Somente as sinapses potenciais acima de um limite são válidas.

Em uma implementação, usamos um número fixo de segmentos dendríticos por célula. Em outra implementação, adicionamos e excluimos segmentos durante o treinamento. Ambos os métodos podem funcionar. Se tivermos um número fixo de segmentos dendríticos por célula, será possível armazenar diferentes conjuntos de sinapses no mesmo segmento. Por exemplo, digamos que temos 20 sinapses válidas em um segmento e um limite de 15. (Em geral, queremos que o limite seja inferior ao número de sinapses para melhorar a imunidade ao ruído). O segmento pode agora reconhecer um estado específico das células vizinhas. O que aconteceria se nós adicionássemos mais 20 sinapses para o mesmo segmento que representa um estado completamente diferente das células vizinhas? Ele introduzirá a possibilidade de erro pois o segmento poderia adicionar 8 sinapses ativas a partir de um padrão e 7 sinapses ativas a partir de outro e tornarem-se ativas incorretamente. Temos descoberto experimentalmente que até 20 diferentes padrões podem ser armazenados em um segmento antes de ocorrerem erros. Portanto, uma célula HTM com uma dúzia de segmentos dendríticos pode participar em muitas predições diferentes.

Sinapses

As sinapses em uma célula HTM possuem um peso binário. Não há nada no modelo HTM que impeça pesos escalares, porém, devido à utilização de padrões distribuídos esparsos nós ainda não tivemos uma necessidade de utilizar pesos escalares.

No entanto, as sinapses em uma célula HTM possuem um valor escalar chamado "permanência" que é ajustado durante a aprendizagem. Um valor de permanência 0,0 representa uma sinapse potencial que não é válida e não avançou em todos os sentidos para se tornar uma sinapse válida. Um valor de permanência acima de um limite (normalmente 0,2) representa uma sinapse que acaba de conectar, mas poderia facilmente ser não-conectados. Um valor de permanência elevado, por exemplo 0,9, representa uma sinapse que está conectada e não pode ser facilmente desconectada.

O número de sinapses válidas nos segmentos dendríticos proximais e distais de uma célula HTM não é fixo. Ele muda conforme a célula é exposta a padrões. Por exemplo, o número de sinapses válidas nos dendritos distais é dependente da estrutura temporal dos dados. Se não houver padrões temporais persistentes na entrada para a região, então todas as sinapses nos segmentos distais terão valores de permanência baixos e muito poucas sinapses serão válidas. Se houver uma grande parte de estrutura temporal no fluxo de entrada, então iremos encontrar muitas sinapses válidas com alta permanência.

Saída da célula

Uma célula HTM tem duas saídas binárias diferentes: 1) a célula está ativa devido a alimentação de entrada (através do dendrito proximal), e 2) a célula está ativa devido às conexões laterais

(através dos segmentos dendríticos distais). A primeira é chamada de "estado ativo" e a segunda de "estado preditivo".

No diagrama anterior, as duas saídas estão representadas por duas linhas que saem do corpo celular quadrangular. A linha da esquerda é o estado ativo de alimentação, enquanto que a linha reta da direita é o estado preditivo.

Somente o estado ativo de alimentação está conectado a outras células na região, assegurando que as predições serão sempre baseadas na entrada atual (mais o contexto). Nós não queremos fazer predições com base em predições. Se fizéssemos isso, quase todas as células da região estariam no estado preditivo após algumas iterações.

A saída da região é um vetor que representa o estado de todas as células. Esse vetor passa a ser a entrada para a próxima região da hierarquia, se houver. Esta saída é o OU dos estados ativos e preditivos. Ao combinar tanto estado ativo como preditivo, a saída da nossa região será mais estável (mudança lenta) do que a entrada. Esta estabilidade é uma propriedade importante da inferência em uma região.

Sugestão de leitura

Muitas vezes somos convidados a sugerir material de leitura para se aprender mais sobre neurociência. O campo da neurociência é tão grande que uma introdução geral requer procurar em muitas fontes diferentes. Novas descobertas são publicadas em revistas acadêmicas que são difíceis de ler e difíceis de obter acesso caso você não tenha uma afiliação universitária.

Aqui estão dois livros disponíveis que um leitor dedicado pode querer dar uma olhada e que são relevantes para os tópicos neste apêndice.

Stuart, Greg, Spruston, Nelson, Häusser, Michael, *Dendritos, segunda edição* (New York: Oxford University Press, 2008)

Este livro é uma boa fonte de tudo sobre dendritos. O capítulo 16 discute as propriedades não-lineares dos segmentos dendríticos utilizados nos algoritmos de aprendizagem cortical HTM. Foi escrito por Mel Bartlett, que contribuiu grandemente para o pensamento neste campo.

Mountcastle, Vernon B. *Neurociência da Percepção: O Córtex Cerebral* (Cambridge, Mass.: Harvard University Press, 1998)

Este livro é uma boa introdução a tudo sobre o neocórtex. Vários dos capítulos discutem os tipos de células e suas conexões. Você poderá ter um boa noção dos neurônios corticais e suas conexões, embora seja antigo demais para cobrir os conhecimentos mais recentes das propriedades dendríticas.

Apêndice B: Uma Comparação das Camadas no Neocórtex e em uma Região HTM

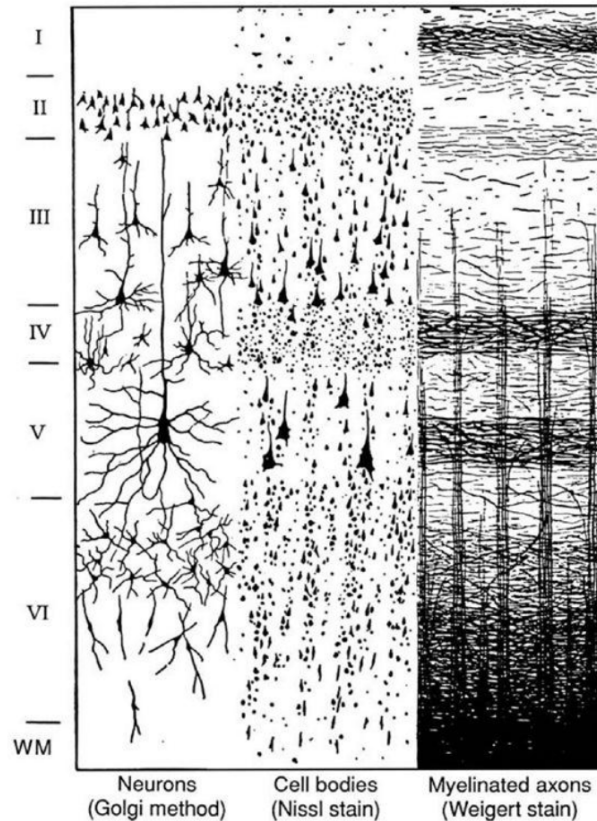
Este apêndice descreve a relação entre uma região HTM e uma região do neocórtex biológico.

Especificamente, o apêndice aborda como o algoritmo de aprendizagem cortical HTM, com suas colunas e células, referencia a arquitetura em camadas e colunar do neocórtex. Muitas pessoas ficam confusas com o conceito de "camadas" no neocórtex e como ele está relacionado com uma camada HTM. Esperamos que este apêndice resolva a confusão, bem como forneça mais detalhes sobre a biologia subjacente ao algoritmo de aprendizagem cortical HTM.

Circuitos do neocórtex

O neocórtex humano é uma folha de tecido neural com aproximadamente 1.000 cm² de área e 2 mm de espessura. Para visualizar esta folha, pense em um guardanapo de pano, que é uma aproximação razoável da área e da espessura do neocórtex. O neocórtex é dividido em dezenas de regiões funcionais, algumas relacionadas com a visão, outras com a audição, outras com a linguagem, e assim por diante. Vistas sob um microscópio, as características físicas das diferentes regiões parecem notavelmente similares.

Existem vários princípios de organização observados em cada região ao longo do neocórtex.



Camadas

O neocórtex é geralmente expresso em seis camadas. Cinco delas contêm células e uma contém mais conexões. As camadas foram descobertas há mais de cem anos com o advento de técnicas de coloração. A imagem acima (de Cajal) mostra uma pequena fatia do neocórtex exposta usando três diferentes métodos de coloração. O eixo vertical mede a espessura do neocórtex, aproximadamente 2mm. O lado esquerdo da imagem indica as seis camadas. A camada 1, no topo, é o nível não-celular. O "WM" na parte inferior indica o início da substância branca, onde os axônios das células viajam para outras partes do neocórtex e outras partes do cérebro.

O lado direito da imagem é uma mancha que mostra apenas os axônios mielinizados. (Mielinação é uma bainha de gordura que cobre alguns, mas não todos, os axônios). Nesta parte da imagem você pode ver dois dos principais princípios de organização do neocórtex, camadas e colunas. A maioria dos axônios se dividem em dois imediatamente após deixar o corpo do neurônio. Um ramo viajará principalmente na horizontal e outro ramo viajará principalmente na vertical. O ramo horizontal faz um grande número de conexões com outras células na mesma camada ou nas proximidades, assim, as camadas se tornam visíveis em manchas como esta. Tenha em mente que este é um desenho de uma fatia do neocórtex. A maioria dos axônios estão entrando e saindo do plano da imagem, portanto, os axônios são mais longos do que aparecem na imagem. Estima-se que exista entre 2 à 4 km de axônios e dendritos em cada milímetro cúbico do neocórtex.

A seção do meio da imagem é uma mancha que mostra corpos de neurônios, mas não mostra quaisquer dendritos ou axônios. Você pode ver que os tamanhos e as densidades dos neurônios também variam por camada. Existe apenas uma pequena indicação de colunas nesta imagem em particular. Você pode notar que existem alguns neurônios na camada 1. O número de neurônios da camada 1 é tão pequeno que a camada chega a ser referida como uma camada não-celular.

Neuro-cientistas estimaram que exista algo em torno de 100 mil neurônios em um milímetro cúbico do neocórtex.

A parte esquerda da imagem é uma mancha que mostra o corpo, os axônios e os dendritos de apenas alguns neurônios. Você pode ver que os tamanhos das "árvores" de dendrito variam significativamente nas células de camadas diferentes. Também são visíveis alguns "dendritos apicais" que emergem do corpo celular para fazer conexões em outras camadas. A presença e o destino dos dendritos apicais são específicos para cada camada.

Em suma, a organização colunar e em camadas do neocórtex se torna evidente quando o tecido neural é manchado e visto sob um microscópio.

Variações das camadas em diferentes regiões

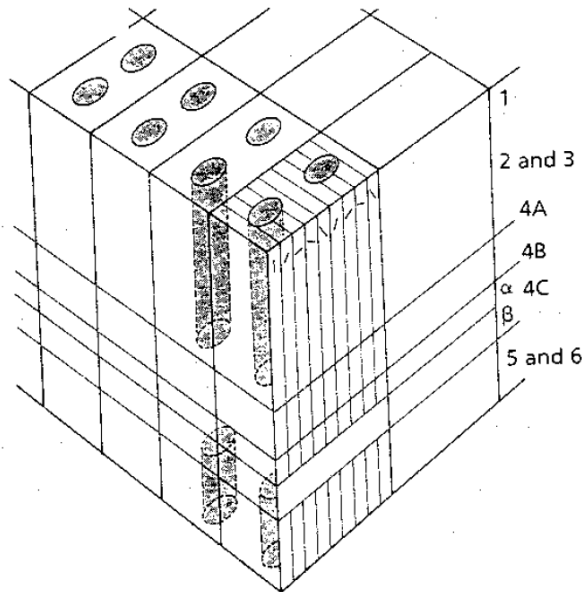
Existe uma variação na espessura das camadas de diferentes regiões do neocórtex e algumas discordâncias sobre o número de camadas. As variações dependem de qual animal está sendo estudado, qual região está sendo olhada, e quem está olhando. Por exemplo, na imagem acima, a camada 2 e a camada 3 aparecem facilmente distinguidas, mas geralmente isso não é o caso. Alguns cientistas relatam que não conseguem distinguir as duas camadas nas regiões que estudam, portanto, muitas vezes as camadas 2 e 3 são agrupadas juntas e chamadas de "camada 2/3". Outros cientistas vão na direção oposta, definindo sub-camadas tais como 3A e 3B.

A camada 4 é a mais bem definida nas regiões neocorticais que estão mais perto dos órgãos sensoriais. Enquanto que em alguns animais (os seres humanos e macacos, por exemplo), a camada 4 na primeira região da visão é claramente subdividida. Em outros animais não é. A camada 4 na maior parte desaparece nas regiões hierarquicamente longe dos órgãos sensoriais.

Colunas

O segundo princípio organizador mais importante do neocórtex é colunas. Alguma organização colunar é visível em imagens coradas, mas a maioria das evidências para as colunas está baseada em como as células respondem às diferentes entradas.

Quando os cientistas utilizam sondas para ver o que faz com que os neurônios se tornem ativos, eles descobrem que os neurônios que estão alinhados verticalmente, cruzando camadas diferentes, respondem de forma aproximada a mesma entrada.



Este desenho ilustra algumas das propriedades de resposta das células em V1, a primeira região cortical que processa a informação da retina.

Uma das primeiras descobertas foi que a maioria das células em V1 responde às linhas ou bordas de diferentes orientações em áreas específicas da retina. As células que estão alinhadas verticalmente em colunas respondem a todas as arestas com a mesma orientação. Se você olhar cuidadosamente, verá que o desenho mostra um conjunto de pequenas linhas em diferentes orientações dispostas na parte superior da seção. Essas linhas indicam quais células de orientação de linha naquele local respondem. As células que estão alinhadas verticalmente (dentro das finas listras verticais) respondem às linhas da mesma orientação.

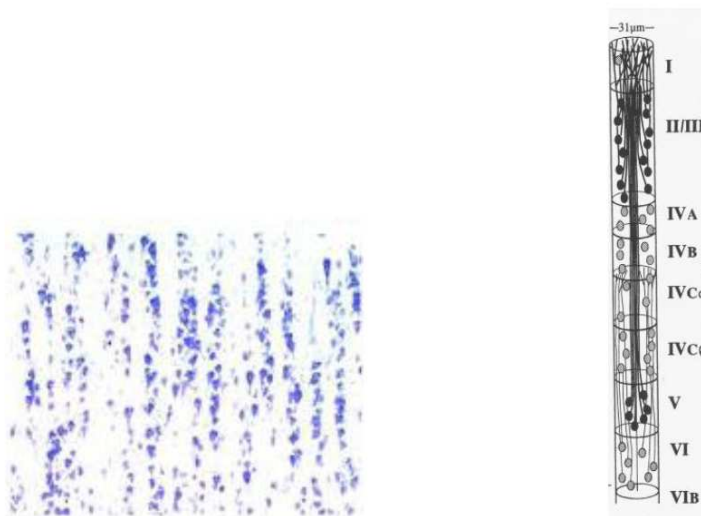
Existem várias outras propriedades colunares vistas em V1, duas das quais são mostradas no desenho. Há "colunas de dominância ocular" onde as células respondem a combinações similares de influência do olho esquerdo e direito. E há "bolhas" onde as células são essencialmente sensíveis a cores. As colunas de dominância ocular são os grandes blocos no diagrama. Cada coluna de dominância ocular inclui um conjunto de colunas de orientação. As "bolhas" são os ovais escuros.

A regra geral para o neocórtex é que várias diferentes propriedades de resposta estão sobrepostas umas sobre as outras, tais como a orientação e a dominância ocular. A medida que você se move horizontalmente na superfície cortical, a combinação das propriedades de resposta exibidas pelas células muda. No entanto, neurônios alinhados verticalmente compartilham o mesmo conjunto de propriedades de resposta. Este alinhamento vertical é real nas áreas auditiva, visual e somatossensorial. Existem algumas discussões entre os neurocientistas sobre se isso é real em todo o neocórtex, mas parece ser real na maioria das áreas, se não todas.

Mini-colunas

A menor estrutura colunar no neocórtex é a mini-coluna. Mini-colunas possuem aproximadamente 30 microns de diâmetro e contêm de 80 à 100 neurônios cruzando todas as cinco camadas celulares. Todo o neocórtex é composto de mini-colunas. Você pode visualizá-las como pequenos pedaços de espaguete empilhadas lado a lado. Existem intervalos minúsculos

com poucas células entre as mini-colunas, por vezes, tornando-as visíveis em imagens manchadas.



À esquerda está uma imagem manchada que mostra corpos celulares de neurônios na parte de uma fatia neocortical. A estrutura vertical de mini-colunas é evidente nesta imagem. À direita está um desenho conceitual de um mini-coluna (de Peters e Yilmaz). Na realidade é mais fino do que isso. Note que existem vários neurônios em cada camada da coluna. Todos os neurônios em uma mini-coluna irão responder à entradas semelhantes. Por exemplo, no desenho de uma seção de V1 mostrado anteriormente, uma mini-coluna irá conter células que responderão a linhas de uma determinada orientação com uma preferência de dominância ocular específica. As células em uma mini-coluna adjacente poderão responder a uma orientação de linha um pouco diferente ou a uma diferente preferência de dominância ocular.

Os neurônios inibitórios desempenham um papel essencial que é o de definir mini-colunas. Eles não são visíveis na imagem ou no desenho, porém eles enviam axônios em um caminho direto entre mini-colunas dando-lhes parcialmente sua separação física. Os neurônios inibitórios são também creditados por ajudar a forçar todas as células na mini-coluna a responderem à entradas semelhantes.

A mini-coluna é o protótipo para a coluna usada no algoritmo de aprendizagem cortical HTM.

Uma exceção às respostas colunares

Existe uma única exceção às respostas colunares que é relevante para os algoritmos de aprendizagem cortical HTM. Normalmente os cientistas acham que uma célula responde pela exposição de um animal experimental a um estímulo simples. Por exemplo, eles podem mostrar a um animal uma única linha em uma pequena parte do espaço visual para determinar as propriedades de resposta das células em V1. Ao usarem entradas simples, pesquisadores descobriram que as células respondiam sempre a uma mesma entrada. No entanto, se a entrada simples fosse incorporada em um vídeo de uma cena natural, as células se tornavam mais seletivas. Uma célula que responde de forma confiável a uma linha vertical isolada nem sempre responde quando a linha vertical é incorporada em uma imagem complexa se movendo em uma cena natural.

No algoritmo de aprendizagem cortical HTM, todas as células HTM em uma coluna compartilham as mesmas propriedades de alimentação, mas em uma sequência temporal

aprendida, apenas uma das células em uma coluna HTM se torna ativa. Esse mecanismo é o meio de representar seqüências de ordem variável e é análogo à propriedade que acabamos de descrever para os neurônios. Uma entrada simples sem contexto fará com que todas as células em uma coluna se tornem ativas. A mesma entrada em uma seqüência aprendida fará com que apenas uma célula se torne ativa.

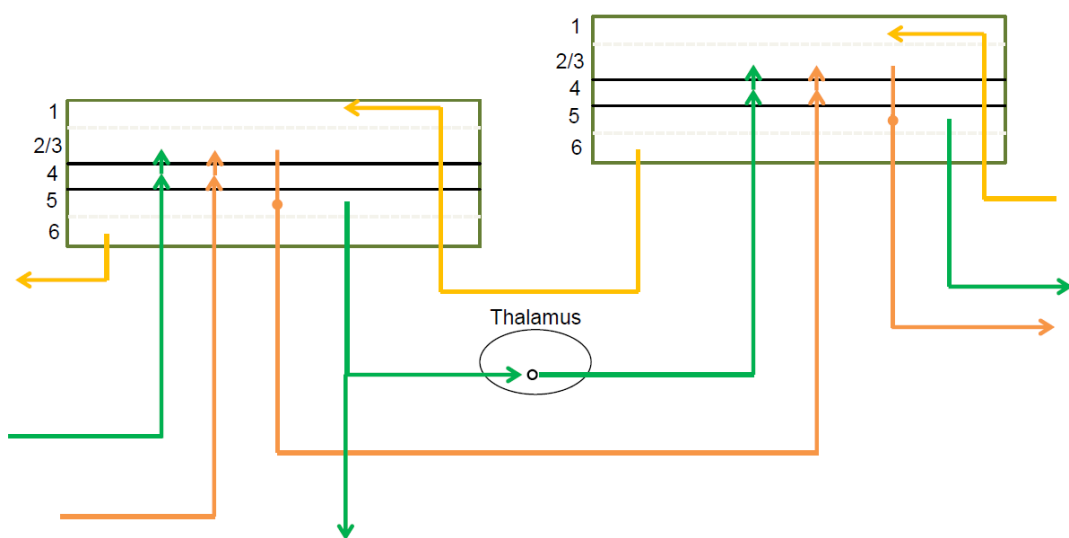
Nós não estamos sugerindo que apenas um neurônio dentro de uma mini-coluna estará ativo por vez. O algoritmo de aprendizagem cortical HTM sugere que dentro de uma coluna, todos os neurônios em uma camada estariam ativos para uma entrada inesperada e um subconjunto de neurônios estariam ativos para uma entrada antecipada.

Por que existem camadas e colunas?

Ninguém sabe ao certo por que existem camadas e porque existem colunas no neocórtex. A teoria HTM, no entanto, propõe uma resposta. O algoritmo de aprendizagem cortical HTM mostra que uma camada de células organizadas em colunas pode ser uma memória de alta capacidade de transições de estado de ordem variável. De forma mais simples, uma camada de células pode aprender um monte de seqüências. As colunas de células que compartilham a mesma resposta de alimentação são o mecanismo chave para a aprendizagem de transições de ordem variável.

Essa hipótese explica por que as colunas são necessárias, mas e o que dizer das cinco camadas? Se uma única camada cortical pode aprender seqüências e fazer predições, porque nós vemos cinco camadas no neocórtex?

Nós propomos que as diferentes camadas observadas no neocórtex estão todas aprendendo seqüências utilizando o mesmo mecanismo básico, porém as seqüências aprendidas em cada camada são utilizadas de diferentes maneiras. Existe muita coisa que não entendemos sobre isso, mas podemos descrever a idéia geral. Antes de fazermos isto, será útil descrever como os neurônios em cada camada se conectam.



O diagrama acima ilustra duas regiões neocorticais e as principais conexões entre elas. Essas conexões são vistas em todo o neocórtex onde duas regiões se projetam uma à outra. O quadro à

esquerda representa uma região cortical que é hierarquicamente inferior à região (quadro) à direita, portanto as informações de alimentação vão da esquerda para a direita no diagrama. A seta para baixo se projeta para outras áreas do cérebro. Informações de realimentação vão da direita para a esquerda. Cada região é dividida em camadas. As camadas 2 e 3 são mostradas como camada 2/3.

As linhas coloridas representam a saída dos neurônios nas diferentes camadas. Estas são feixes de axônios provenientes dos neurônios na camada. Lembre-se que os axônios são imediatamente divididos em dois. Um ramo se espalha horizontalmente na região, principalmente dentro da mesma camada. Assim, todas as células em cada camada estão fortemente interconectadas. Os neurônios e as conexões horizontais não são mostrados no diagrama.

Existem dois caminhos de alimentação, um caminho direto mostrado em laranja e um caminho indireto mostrado em verde. A camada 4 é a principal camada de entrada de alimentação e recebe a entrada de ambas as vias de alimentação. A camada 4 se projeta para a camada 3.

A camada 3 também é origem da via de alimentação direta. Então, o caminho direto de alimentação está limitado à camada 4 e à camada 3.

Algumas conexões de alimentação pulam a camada 4 e vão diretamente para a camada 3. E, como mencionado acima, a camada 4 desaparece em regiões distantes da entrada sensorial. Nesse ponto, o caminho direto de alimentação simplesmente parte de camada 3 para camada 3 na próxima região.

A segunda via de alimentação (mostrada em verde) se origina na camada 5. As células da camada 3 fazem uma conexão com as células da camada 5 à medida que passam em seu caminho para a próxima região. Depois de sair da folha cortical, os axônios das células da camada 5 se dividem novamente. Um ramo se projeta para áreas sub-corticais do cérebro que estão envolvidas na geração motora. Esses axônios são creditados para serem os comandos motores (como mostrado na seta indicando para baixo). O outro ramo se projeta para uma parte do cérebro chamada de tálamo, que atua como um portão. O tálamo por outro lado passa a informação para a próxima região ou a bloqueia.

Finalmente, a principal via de realimentação, mostrada em amarelo, começa na camada 6 e se projeta para a camada 1. As células nas camadas 2, 3 e 5 se conectam a camada 1 através de seus dendritos apicais (não mostrado). A camada 6 recebe entrada da camada 5.

Esta descrição é um resumo limitado do que é conhecido sobre conexões de camada à camada. Mas é suficiente para compreender a nossa hipótese sobre por que existem várias camadas se todas as camadas estão aprendendo seqüências.

Hipótese sobre o que as diferentes camadas fazem

Nós propomos que as camadas 3, 4 e 5 são todas camadas de alimentação e estão todas aprendendo seqüências. A camada 4 aprende seqüências de primeira ordem. A camada 3 aprende seqüências de ordem variável. E a camada 5 aprende seqüências de ordem variável com o tempo. Vejamos cada uma dessas em detalhes.

Camada 4

É fácil de aprender seqüências de primeira ordem usando o algoritmo de aprendizagem cortical HTM. Se não formos as células em uma coluna para inibir uma a outra, isto é, se as células em uma coluna não se diferenciarem no contexto das entradas anteriores, então a aprendizagem de primeira ordem irá ocorrer. No neocórtex isto provavelmente seria realizado através da remoção de um efeito inibitório entre as células na mesma coluna. Em nossos modelos computacionais do algoritmo de aprendizagem cortical HTM, apenas atribuímos uma célula por coluna, o que acaba produzindo um resultado semelhante.

Seqüências de primeira ordem são o que é necessário para formar representações invariáveis das transformações espaciais de uma entrada. Na visão, por exemplo, tradução x-y, escala, e rotação são todas transformações espaciais. Quando uma região HTM com memória de primeira ordem é treinada com objetos em movimento, ela descobre que diferentes padrões espaciais são equivalentes. As células HTM resultantes irão se comportar como o que são chamadas de "células complexas" no neocórtex. As células HTM permanecerão ativas (no estado preditivo) mesmo com várias transformações espaciais.

Na Numenta fizemos experimentos de visão que verificam se este mecanismo funciona como o esperado, e se alguma invariância espacial é alcançada dentro de cada nível. Os detalhes dessas experiências estão além do escopo deste apêndice.

A aprendizagem de seqüências de primeira ordem na camada 4 é consistente com a constatação de células complexas na camada 4, e explica porque a camada 4 desaparece em regiões mais elevadas do neocórtex. A medida que você for subindo na hierarquia em algum momento não será mais possível aprender mais invariâncias espaciais pois as representações já serão invariáveis para elas.

Camada 3

A camada 3 está mais próxima ao algoritmo de aprendizagem cortical HTM que descrevemos no Capítulo 2. Ela aprende seqüências de ordem variável e formas predições que são mais estáveis do que a sua entrada. A camada 3 sempre se projeta para a próxima região na hierarquia, e portanto, conduz ao aumento da estabilidade temporal dentro da hierarquia. A memória de seqüências de ordem variável lida com neurônios chamados de "células complexas direcionalmente afinadas" que são observadas pela primeira vez na camada 3. Células complexas direcionalmente afinadas se diferenciam pelo contexto temporal, tais como uma linha se movendo da esquerda versus uma linha se movendo da direita.

Camada 5

A camada de alimentação final é a camada 5. Nós propomos que a camada 5 é semelhante à camada 3, com três diferenças. A primeira diferença é que a camada 5 adiciona um conceito de tempo. A camada 3 prediz "o que" vai acontecer, mas isso não lhe diz "quando" isso vai acontecer. No entanto, muitas tarefas requerem tempo, tais como o reconhecimento de palavras faladas em que o tempo relativo entre os sons é importante. O comportamento motor é outro exemplo; a sincronização entre as ativações musculares é essencial. Nós propomos que os neurônios da camada de 5 predizem o próximo estado somente depois do momento esperado. Há vários detalhes biológicos que sustentam esta hipótese. Um deles é que a camada 5 é a camada de saída motora do neocórtex. O outro é que a camada 5 recebe entrada de uma camada que se origina em uma parte do tálamo (não mostrado no diagrama). Nós propomos que esta informação diz sobre como o tempo é codificado e distribuído para muitas células através de uma entrada do tálamo para a camada 1 (não mostrado no diagrama).

A segunda diferença entre as camadas 3 e 5 é que queremos que a camada 3 faça previsões como uma medida para o futuro possível, ganhando estabilidade temporal. O algoritmo de aprendizagem cortical HTM descrito no Capítulo 2 faz isso. Em contraste, queremos apenas que a camada 5 prediga o próximo elemento (em um tempo específico). Nós não temos modelado essa diferença, mas ela naturalmente ocorreria se as transições fossem sempre armazenadas com um tempo associado.

A terceira diferença entre as camadas 3 e 5 pode ser vista no diagrama. A saída da camada 5 sempre se projeta para centros motores sub-corticais, e o caminho de alimentação é bloqueado pelo tálamo. A saída da camada 5 às vezes é passada para a próxima região e às vezes é bloqueada. Nós (e outros) propomos que este bloqueio está relacionado com a atenção encoberta (atenção encoberta é quando você atende a uma entrada sem comportamento motor).

Em resumo, a camada 5 combina tempo específico, atenção e comportamento motor. Existem muitos mistérios relacionados à forma como estes atuam juntos. O ponto que quero salientar é que uma variação do algoritmo de aprendizagem cortical HTM pode facilmente incorporar tempo específico e justificar uma camada separada no córtex.

Camadas 2 e 6

A camada 6 é a origem dos axônios que realimentam as regiões mais baixas. Muito pouco se sabe sobre a camada 2. Como mencionado acima, a própria existência da camada 2 como parte da camada 3 às vezes é discutida. Não teremos mais a dizer sobre esta questão agora além de salientar que as camadas 2 e 6, assim como todas as outras camadas, exibem o padrão de conexões horizontais em massa e de propriedades de resposta colunares, por isso propomos que elas também estão executando uma variante do algoritmo de aprendizagem cortical HTM.

O que faz uma região HTM corresponder ao neocórtex?

Temos implementado o algoritmo de aprendizagem cortical HTM em duas versões, uma com várias células por coluna para a memória de ordem variável, e uma com uma única célula por coluna para a memória de primeira ordem. Acreditamos que estes dois tipos correspondem às camadas 3 e 4 no neocórtex. Nós não tentamos combinar essas duas variantes em uma única região HTM.

Embora o algoritmo de aprendizagem cortical HTM (com várias células por coluna) seja o mais próximo à camada 3 no neocórtex, temos uma flexibilidade em nossos modelos que o cérebro não tem. Portanto, podemos criar camadas celulares híbridas que não correspondem a camadas neocorticais específicas. Por exemplo, em nosso modelo sabemos a ordem em que as sinapses são formadas por segmentos dendríticos. Podemos usar essa informação para extrair o que está predito para acontecer a partir da previsão mais geral de todas as coisas que vão acontecer no futuro. Provavelmente, podemos adicionar tempo específico da mesma forma. Portanto, deveria ser possível criar uma região HTM de camada única que combinasse as funções da camada 3 e da camada 5.

Sumário

O algoritmo de aprendizagem cortical HTM encarna o que acreditamos ser um elemento básico de organização neural no neocórtex. Ele mostra como uma camada de neurônios conectados horizontalmente aprende seqüências de representações distribuídas esparsas. Variações do algoritmo de aprendizagem cortical HTM são utilizadas em diferentes camadas do neocórtex relacionadas, mas de diferentes propósitos.

Propomos que a entrada de alimentação para uma região neocortical, caso seja para a camada 4 ou para camada 3, se projeta predominantemente para dendritos proximais, que com o auxílio de células inibitórias, criam uma representação distribuída esparsa da entrada. Propomos que as células nas camadas 2, 3, 4, 5, e 6 compartilham esta representação distribuída esparsa. Isso é realizado por forçar todas as células em uma coluna que abrange as camadas a responder à mesma entrada de alimentação.

Propomos que as células da camada 4, quando presentes, usam o algoritmo de aprendizagem cortical HTM para aprender transições temporais de primeira ordem que fazem representações que são invariantes diante das transformações espaciais. As células da camada 3 usam o algoritmo de aprendizagem cortical HTM para aprender transições temporais de ordem variável e formar representações estáveis que são passadas acima na hierarquia cortical. As células da camada 5 aprendem transições de ordem variável com o tempo. Nós não temos propostas concretas para a camada 2 e a camada 6. No entanto, devido à conectividade horizontal típica nestas camadas é provável que elas, também, estão aprendendo de alguma forma a memorizar seqüências.

Glossário

Notas: As definições aqui captam como os termos são usados neste documento, e pode ter outros significados de uso geral. Termos em letras maiúsculas referem-se a outros termos definidos neste glossário.

Estado Ativo	um estado no qual as Células estão ativas devido à entrada de Alimentação
Baixo-para-Cima	sinônimo para Alimentação
Células	o equivalente HTM de um Neurônio <i>Células são organizadas em colunas nas regiões HTM.</i>
Atividade Coincidente	quando duas ou mais Células estão ativas ao mesmo tempo
Coluna	um grupo de uma ou mais Células que funcionam como uma unidade em uma Região HTM <i>As Células dentro de uma coluna representam a mesma entrada de alimentação, mas em contextos diferentes.</i>
Segmentos Dendríticos	uma unidade de integração de Sinapses associadas com Células e Colunas <i>As HTMs têm dois tipos diferentes de segmentos dendríticos. Um está associado com as conexões laterais de uma célula. Quando o número de sinapses ativas no segmento dendrítico excede um limite, a célula associada entra em estado preditivo. O outro está associado com conexões de alimentação de uma coluna. O número de sinapses ativas é somado de forma a gerar a ativação de alimentação de uma coluna.</i>
Densidade Desejada	percentual desejado de Colunas ativadas devido à entrada de Alimentação para uma Região <i>O percentual só se aplica dentro de um raio que varia de acordo com o fan-out das entradas de alimentação. Ele é "desejado" porque o percentual varia com base em uma entrada específica.</i>
Alimentação	movimento em direção oposta de uma entrada, ou de um Nível inferior para um Nível superior em uma Hierarquia (às vezes chamada de Baixo-para-Cima)
Realimentação	movimento em direção à uma entrada, ou a partir de um Nível superior para um Nível inferior em uma Hierarquia (às vezes chamada de Cima-para-Baixo)

Predição de Primeira Ordem	uma predição com base apenas na entrada atual e não nas entradas anteriores - comparar com Predição de Ordem Variável.
Memória Temporal Hierárquica (HTM)	uma tecnologia que reproduz algumas das funções estruturais e algorítmicas do neocórtex
Hierarquia	uma rede de elementos conectados onde as conexões entre os elementos são identificadas exclusivamente como de Alimentação ou de Realimentação
Algoritmos de Aprendizagem Cortical HTM	o conjunto de funções para o Agrupador Espacial, o Agrupador Temporal, e a aprendizagem e esquecimento que compõem uma Região HTM, também conhecido como Algoritmos de Aprendizagem HTM
Rede HTM	uma Hierarquia de Regiões HTM
Região HTM	a principal unidade de memória e Predição em uma HTM <i>Uma região HTM é composta por uma camada de células altamente interconectadas dispostas em colunas. Uma região HTM tem atualmente uma única camada de células, enquanto que no neocórtex (e no final das contas em HTM), uma região terá várias camadas de células. Quando se refere ao contexto da sua posição em uma hierarquia, uma região pode ser referida como um nível.</i>
Inferência	reconhecer um padrão de entrada espacial e temporal como semelhante aos padrões aprendidos anteriormente
Raio de Inibição	define a área em torno de uma Coluna que ela inibe ativamente
Conexões Laterais	conexões entre as Células dentro da mesma Região
Nível	uma Região HTM no contexto da Hierarquia
Neurônio	uma Célula que processa informações no cérebro <i>Neste documento, nós usamos a palavra neurônio especificamente quando nos referimos a células biológicas, e "células" quando nos referimos à unidade de computação HTM.</i>
Permanência	um valor escalar que indica o estado da conexão de uma Sinapse Potencial <i>Um valor de permanência abaixo de um limite indica que a sinapse não está formada. Um valor de permanência acima do limite indica que a sinapse é válida. A aprendizagem em uma região HTM é feita pela modificação dos valores de</i>

	<i>permanência das sinapses potenciais.</i>
Sinapse Potencial	o subconjunto de todas as Células que poderiam formar Sinapses com um determinado Segmento Dendrítico <i>Apenas um subconjunto de sinapses potenciais será válido em qualquer tempo com base em seu valor de permanência.</i>
Predição	ativação de Células (para um estado preditivo), que provavelmente irão se tornar ativas no futuro próximo devido ao entrada de Alimentação <i>Uma região HTM muitas vezes prediz muitas entradas futuras possíveis ao mesmo tempo.</i>
Campo Receptivo	o conjunto de entradas ao qual uma Coluna ou Célula está conectada <i>Se a entrada para uma região HTM está organizada como uma matriz 2D de bits, então o campo receptivo poderá ser expresso em um raio dentro do espaço de entrada.</i>
Sensor	uma fonte de entradas para uma Rede HTM
Representação Distribuída Esparsa	representação composta por muitos bits nos quais uma pequena percentagem está ativa e onde nenhum único bit é suficiente para transmitir um significado
Agrupamento Espacial	o processo de formação de uma representação distribuída esparsa de uma entrada <i>Uma das propriedades do agrupamento espacial é que os padrões de entrada sobrepostos mapeiam para as mesmas representações distribuídas esparsas.</i>
Sub-Amostragem	reconhecer um padrão grande distribuído através da combinação de apenas um pequeno subconjunto de bits ativos no mesmo
Sinapse	conexão formada entre as Células durante a aprendizagem
Agrupamento Temporal	o processo de formação de uma representação de uma seqüência de padrões de entrada onde a representação resultante é mais estável do que a entrada
Cima-para-Baixo	sinônimo para Realimentação
Predição de Ordem Variável	uma predição com base em porções variáveis do contexto anterior - comparar com Predição de Primeira Ordem <i>É chamado de "variável", porque a memória que mantém o contexto anterior é alocada conforme necessário. Assim, um</i>

sistema de memória que implementa predição de ordem variável pode usar o contexto voltando no tempo sem exigir quantidades exponenciais de memória.
