Numenta®

MEMOIRE TEMPORELLE HIERARCHIQUE

Incluant les

Algorithmes d'apprentissage corticaux HTM

VERSION 0.2.1, 12 SEPTEMBRE 2011

Note du Traducteur (NdT)

En accord avec Numenta, nous donnons dans ce préambule la traduction française des termes *Hierarchical Temporal Memory* et *Sparse Distributed Representation*. Dans la suite de l'ouvrage nous avons pris le parti d'utiliser les acronymes anglais HTM et SDR en lieu et place des traductions françaises notamment pour améliorer l'indexation de ce document par les moteurs de recherche.

Hierarchical Temporal Memory : Mémoire Temporelle Hiérarchique (HTM)

Sparse Distributed Representation : Représentation Parcimonieuse Distribuée (SDR)

La traduction de SDR s'appuie sur la terminologie française employée notamment dans les travaux sur la théorie du codage neural de Claude Berrou et Vincent Gripon (voir par exemple l'ouvrage Petite Mathématique du Cerveau paru chez Odile Jacob). La notion de parcimonie rend parfaitement compte de l'économie de moyens (en capacité de stockage et en énergie de fonctionnement) mise en œuvre dans les algorithmes HTM et dans le cortex cérébral.

Pour la traduction des autres termes techniques nous vous invitons à consulter le glossaire situé à la fin du document.

Traduction française par Laurent Julliard <laurent_a t_moldus_d o t_org> / @lrjay

A lire avant toute chose !

Ce document est une version préliminaire. Il y manque un certain nombre de choses dont vous devez avoir connaissance.

Ce qui figure dans ce document:

Ce document décrit en détail les nouveaux algorithmes d'apprentissage et de prédiction développés par Numenta. Le niveau de détail est suffisant pour qu'un programmeur les comprenne et puisse les implémenter s'il le souhaite. Le document débute par un chapitre d'introduction. Si vous avez suivi Numenta et que vous avez lu certains de nos livres blancs, cette introduction vous paraîtra familière. Le reste du document est nouveau.

Ce qui ne figure PAS dans ce document:

Plusieurs points liés à l'implémentation de ces nouveaux algorithmes n'ont pas été traités dans cette version préliminaire.

- Bien que la plupart des aspects de ces algorithmes aient été mis en œuvre et testé, aucun des résultats produits par ces tests n'est actuellement fourni.

- L'application de ces algorithmes à des cas pratiques n'est pas non plus traitée. Il manque une description sur la façon de convertir des données en provenance d'un capteur ou d'une base de données en une représentation distribuée adaptée aux algorithmes.

- Les algorithmes sont capables d'apprendre à la volée mais certains détails nécessaires à l'implémentation de cet apprentissage ne sont pas décrits pour quelques cas particuliers.

- D'autres ajouts sont prévus tels que les propriétés des représentations parcimonieuses distribuées (sparse distributed representations), la description d'applications et d'exemples et les citations en annexe.

Nous mettons ce document à disposition dans son état actuel car nous pensons que ces algorithmes susciteront un intérêt. Les points non traités ne devraient entraver ni la compréhension ni l'expérimentation des algorithmes par des chercheurs motivés. Nous mettrons ce document à jour régulièrement au fur et à mesure de nos avancées.

Table des matières

Préface

Il y a beaucoup de choses que les êtres humains trouvent faciles à faire et que les ordinateurs sont encore aujourd'hui incapables d'accomplir. Des tâches telles que la reconnaissance de formes, la compréhension d'une langue, la reconnaissance et la manipulation d'objets au toucher ou se déplacer dans un monde complexe sont évidentes pour des humains. En dépit de dizaines d'année de recherche, nous disposons de très peu d'algorithmes permettant aux ordinateurs de se mesurer aux capacités de l'être humain.

Chez les Hommes[1] ces aptitudes sont en grande partie du ressort du cortex cérébral. La Mémoire Temporelle Hiérarchique (HTM) est une technologie inspirée de la façon dont le cortex exécute ces fonctions. La HTM devrait permettre de concevoir des machines qui approchent ou dépassent le niveau de performance des Hommes pour de nombreuses tâches cognitives.

Ce document décrit la technologie HTM. Le chapitre 1 en donne un aperçu en soulignant l'importance de l'organisation hiérarchique, les représentations distribuées parcimonieuses et les transitions d'apprentissage basées sur le temps. Le chapitre 2 décrit en détail les algorithmes d'apprentissage cortical. Les chapitres 3 et 4 fournissent le pseudocode pour les algorithmes d'apprentissage HTM qui sont au nombre de deux : le concentrateur spatial (spatial pooler) et le concentrateur temporelle (temporal pooler). Après avoir lu les chapitres 2 à 4, les ingénieurs informaticiens chevronnés devraient être en mesure de reproduire les algorithmes et de les expérimenter. Avec un peu de chance, quelques lecteurs iront plus loin et prolongeront notre travail.

Public visé

Ce document s'adresse à un public ayant des connaissances techniques. Bien que la connaissance des neurosciences ne soit pas un prérequis, nous supposons que vous êtes en mesure de comprendre des concepts mathématiques et informatiques. Nous avons écrit ce document de façon à ce qu'il puisse être donné en lecture dans un cours. Les principales cibles imaginées sont les étudiants en informatique ou en sciences cognitives ou bien les développeurs intéressés par l'élaboration de systèmes cognitifs artificiels et intelligents inspirés des principes de fonctionnement du cerveau humain.

Les lecteurs n'ayant pas une formation technique pourront néanmoins tirer parti de certaines sections du document, en particulier le *Chapitre 1: Aperçu de HTM*.

[11] Par Homme (avec un H majuscule) nous entendons ici l'espèce humaine sans distinction de sexe.

Liens avec les documents antérieurs

Une partie de la théorie qui sous-tend la HTM est décrite dans le livre *On Intelligence* écrit en 2004, dans divers livres blancs publiés par Numenta et dans des articles écrits par les employés de Numenta et revus par la communauté scientifique. Il n'est pas nécessaire d'avoir lu ces documents car l'essentiel se trouve condensé dans ce document. Il faut noter que les algorithmes d'apprentissage HTM décrits dans les chapitres 2 à 4 n'ont jamais été publiés auparavant. Ces nouveaux algorithmes remplacent nos algorithmes de première génération appelés Zeta 1. Pendant quelques temps, nous avons appelé ces nouveaux algorithmes « Représentations distribuées à densité fixe » (Fixed-density Distributed Representations) ou « FDR » mais cette appellation n'a plus cours. Le nouveau nom utilisé est Algorithmes d'Apprentissage Cortical ou parfois, pour condenser, Algorithmes d'Apprentissage HTM.

Nous vous invitons à lire l'ouvrage *On Intelligence*, écrit par le co-fondateur de Numenta Jeff Hawkins en collaboration avec Sandra Blakeslee. Bien que l'ouvrage ne fasse pas mention de la théorie HTM sous ce nom, il en donne une lecture non technique et facile à aborder y compris sur les principes de neurosciences sous-jacents. Lorsque *On Intelligence* a été écrit, nous comprenions ce principes fondateurs mais nous ne savions pas encore comment les décliner sous forme d'algorithmes. Vous pouvez voir ce document comme la suite du travail entamé dans *On Intelligence.*

À propos de Numenta

Numenta, Inc. (www.numenta.com) a été créée en 2005 pour développer la technologie HTM à la fois pour des usages commerciaux et scientifiques. C'est pour atteindre cet objectif que nous documentons en totalité nos avancées et nos découvertes. Nous publions également notre logiciel sous une forme utilisable à des fins de recherche ou d'activité commerciale. Nous l'avons structuré de façon à encourager l'émergence d'une communauté indépendante capable de développer des applications. L'utilisation des logiciels et de la propriété intellectuelle de Numenta, sont libres de droits pour tout usage à des fins de recherche. Nous génèrerons des revenus en proposant du support, des licences de notre logiciel et de notre propriété intellectuelle pour des déploiements commerciaux. Nous rechercherons toujours le moyen d'amener nos partenaires vers le succès tout en réussissant nous-mêmes.

Numenta est une société privée basée à Redwood City, Californie, USA.

A propos des auteurs

Ce document est le résultat d'un effort conjoint des employés de Numenta. Les noms des auteurs principaux de chaque section sont mentionnés dans l'historique du document.

Historique du document

Nous avons porté dans la table ci-dessous les principaux changements intervenus entre les versions. Les modifications mineures telles que clarifications ou reformatages n'y figurent pas.

Version	Date	Changes	Principal Authors
0.1	9 Nov 2010	1. Préface, Chapitres 1, 2, 3, 4 et Glossaire : première version	Jeff Hawkins, Subutai Ahmad, Donna Dubinsky
0.1.1	23 Nov 2010	1. Chapitre 1: clarification de la terminologie telle que niveaux, colonnes et couches dans la section Régions, 2. Annexe A: première version	Hawkins & Dubinsky Hawkins
0.2	10 Déc 2010	1. Chapitre 2: diverses clarifications 2. Chapitre 4: mise à jour des lignes références, modification du code en ligne 37 et 39 3. Annexe B: première version	Hawkins Ahmad Hawkins
0.2.1	12 Sep 2011	1. À lire avant toute chose: référence à 2010 supprimée 2. Préface: Suppression de la section sur les versions	

Chapitre 1 : Tour d'horizon des MTHs

La Mémoire Temporelle Hiérarchique (HTM) est une technologie d'apprentissage automatique qui vise à exploiter les propriétés structurelles et algorithmiques du cortex cérébral.

Le cortex est le siège de l'intelligence dans le cerveau des mammifères. La vision à haut niveau, l'ouïe, le toucher, le mouvement, le langage et la planification sont tous du ressort du cortex. Etant donné la diversité des ce fonctions cognitives, vous pourriez vous attendre à ce que le cortex mettre en œuvre une série tout aussi variée d'algorithmes neurales spécifiques à chaque fonction. Il n'en est rien. Le cortex montre en effet une circuiterie neuronales d'une remarquable uniformité. Les données biologiques suggèrent que le cortex met en œuvre un ensemble d'algorithmes communs pour effectuer de nombreuses fonctions intelligentes toutes différentes.

HTM fournit un cadre théorique pour comprendre le cortex et ses nombreuses capacités. À ce jour nous avons implémenté un sous-ensemble restreint de ce cadre théorique. Au fil du temps nous complèterons l'implémentation. Aujourd'hui nous pensons avoir implémenté un sous-ensemble suffisamment important des fonctions du cortex pour que ce travail ait une valeur commerciale et scientifique.

La programmation des MTHs ne ressemble pas à la programmation classique. Avec les ordinateurs d'aujourd'hui, les programmeurs créent des programmes spécifiques pour résoudre des problèmes spécifiques. A l'opposé, les MTHs sont entrainés par le biais de flux de données sensorielles. Les capacités d'une HTM dépendent en grande partie de ce à quoi elle a été exposée.

Les MTHs peuvent être vues comme un type de réseau de neurones. Par définition, tout système qui essaie de modéliser les détails architecturaux du cortex est un réseau de neurones. Toutefois, en lui-même, le terme "réseau de neurones" n'est pas très utile car il s'applique à une large variété de systèmes. Les MTHs modélisent les neurones (appelés cellules dans le langage HTM) en les arrangeant en colonnes, en couches, en régions puis en hiérarchie. Ces précisions sont importantes et de ce point de vue les MTHs sont une nouvelle forme de réseau de neurones.

Comme le laisse entendre le nom, une HTM est fondamentalement un système basé sur une mémoire. Les réseaux de HTM sont entrainés sur un grand nombre de données variables dans le temps et s'appuient sur le stockage d'un ensemble très large de patrons et de séquences. La façon dont les données sont stockées et relues est différente des modèles standards utilisés aujourd'hui par les programmeurs. Les mémoires des ordinateurs classiques ont une organisation à plat sans aucune notion du temps. Un programmeur peut donc implémenter n'importe quel type d'organisation et de structure de données au-dessus des mémoires actuelles. Ils contrôlent comment et où les données sont stockées. A l'opposé une HTM est plus

restrictive. Elle possède une structure et une organisation qui sont intrinsèquement liées au temps. L'information est toujours stockée de façon distribuée. Un utilisateur de HTM spécifie la taille de la hiérarchie et sur quoi entrainer le système mais la HTM contrôle où et comment les informations sont stockées.

Bien que les réseaux HTM se différencient de la programmation classique, nous pouvons utiliser des ordinateurs conventionnels pour les modéliser en intégrant les fonctions clés de la hiérarchie, du temps et la représentation parcimonieuse distribuée (décrits plus loin). Nous pensons que dans le futur, des circuits spécialisés permettant de construire des réseaux MTHs seront disponibles.

Dans ce document, nous illustrons souvent les propriétés et les principes des MTHs à partir d'exemples tirés de la vision, du toucher, de l'ouïe, du langage ou du comportement humain. Ces exemples sont utiles car intuitifs et faciles à appréhender. Toutefois, il est important de garder à l'esprit que les capacités des MTHs sont très larges. On peut tout aussi bien les exposer à des flots de données qui ne proviennent pas des sens humains comme des signaux radars ou infrarouge ou des flux d'informations pures telles que les données des marchés financiers, les données météorologiques, du trafic Web ou du texte. Les MTHs sont des machines à apprendre et à prédire qui peuvent s'appliquer à toutes sortes de problèmes.

Principes des MTHs

Dans cette section, nous traiterons de quelques-uns des principes fondateurs des MTHs : pourquoi leur organisation hiérarchique est importante, comment les régions sont structurées, pourquoi les données sont stockées selon une représentation parcimonieuse distribuée et pourquoi l'évolution temporelle de l'information est critique.

Hiérarchie

Un réseau HTM est composé de régions organisées de façon hiérarchique. La région est l'unité principale de mémoire et de prédiction et nous la détaillerons dans la prochaine section. Chaque région HTM représente typiquement un niveau de hiérarchie. En montant dans la hiérarchie il y a toujours convergence de l'information c'est-à-dire que plusieurs éléments émanant de régions filles convergent vers un seul élément d'une région parent. Cependant, en raison de connexions de rétroaction, l'information diverge aussi en descendant la hiérarchie (une région et un niveau sont quasiment synonymes. Nous utilisons le mot « région » lorsque nous évoquons les fonctions internes d'une région et le mot « niveau » pour faire référence au rôle que remplit la région dans la hiérarchie).

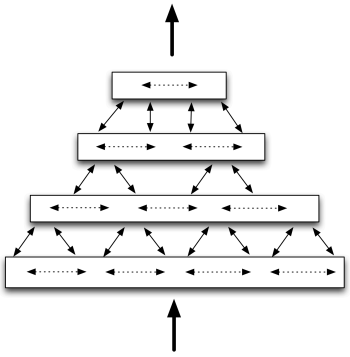


Figure 1.1 : Diagramme simplifié de quatre régions HTM organisée en une hiérarchie à 4 niveaux où l'information circule au sein d'un niveau, entre les niveaux et de/vers l'extérieur de la hiérarchie.

Il est possible de combiner plusieurs réseaux HTM si vous avez plus d'une source de données ou plus d'un capteur. Un réseau pourrait par exemple traiter une information auditive et un autre une information visuelle. Le phénomène de convergence de l'information se produit dans chaque réseau, les branches séparées finissant par converger uniquement au niveau le plus haut.

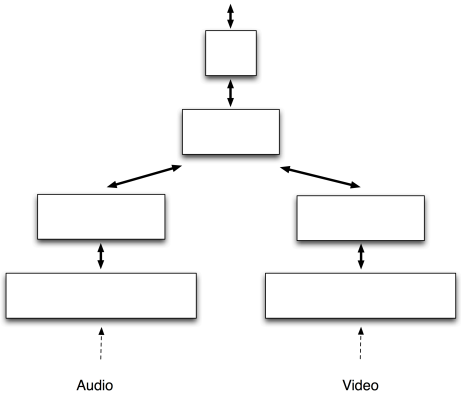


Figure 1.2 : Réseaux convergents de plusieurs capteurs

Le bénéfice de l'organisation hiérarchique réside dans son efficacité. Elle réduit de façon importante le temps d'apprentissage et d'utilisation de la mémoire car les motifs appris à chaque niveau de la hiérarchie sont réutilisés pour être combinés de façons nouvelles dans les niveaux supérieurs. À titre d'illustration, considérons le cas de la vision. Au niveau le plus bas de la hiérarchie, votre cerveau stocke des informations concernant de toutes petites surfaces du champ visuel comme des arêtes ou des coins. Une arête est un composant visuel fondamental de très nombreux objets dans ce monde. Ces motifs de bas niveau sont ensuite recombinés aux niveaux intermédiaires en des composants plus complexes tels que des courbes

et des textures. Un arc peut représenter le bord d'une oreille, la partie supérieure d'un volant ou le bord d'une tasse. Ces motifs intermédiaires sont eux-mêmes recombinés pour représenter des caractéristiques de plus haut niveau comme des visages, des voitures ou des maisons. Pour apprendre un nouvel objet de haut niveau vous n'avez pas à réapprendre tous les composants visuels de plus bas niveau.

De la même façon, lorsque vous apprenez un mot vous n'avez pas besoin de réapprendre à chaque fois les lettres, syllabes et phonèmes qui le composent.

La mise en commun de ces représentations au sein d'une hiérarchie amène aussi des capacités de généralisation sur les comportements attendus. Ainsi lorsque vous voyez un animal avec une bouche et des dents vous vous attendez à ce que l'animal mange avec sa bouche et qu'il puisse éventuellement vous mordre. La hiérarchie permet à un nouvel objet d'hériter des propriétés connues de ses sous-composants.

Combien de choses un seul niveau de hiérarchie HTM peut-il apprendre ? Ou dit autrement, combien de niveaux faut-il pour un apprentissage donné ? Il existe un compromis entre la quantité de mémoire allouée à chaque niveau et le nombre de niveaux nécessaires. Par chance, les MTHs apprennent automatiquement la meilleure représentation possible à chaque niveau étant données la nature des informations en entrée et la quantité de ressources allouées. Si vous allouez davantage de mémoire à un niveau, celui-ci formera des représentations plus grandes et plus complexes ce qui signifie que la hiérarchie pourrait avoir besoin de moins de niveaux. Si au contraire vous allouez moins de mémoire à un niveau, il formera des représentations plus petites et plus simples nécessitant ensuite davantage de niveaux hiérarchiques.

Jusqu'ici nous avons décrit des problèmes difficiles tels que la reconnaissance visuelle. Mais de nombreux problèmes sont plus simples que la vision et peuvent se satisfaire d'une seule région HTM. Nous avons par exemple développé une HTM permettant de prédire où le visiteur d'un site Web va effectuer son prochain clic. Ce problème nécessite d'alimenter le réseau HTM avec un flux de données composés de clics Web. Il ne nécessite que peu ou pas de hiérarchie spatiale puisque la solution passe par la découverte de statistiques temporelles permettant de prédire où l'utilisateur va effectuer son prochain clic étant donné une séquence typique de clics précédents. Les algorithmes d'apprentissage temporel des HTM sont idéals pour ce type de problèmes.

En résumé, les hiérarchies de HTM réduisent le temps d'entrainement, l'empreinte mémoire et introduisent une forme de généralisation. Par ailleurs, nombre de problèmes de prédiction simples peuvent être résolus avec une seule région HTM.

Les régions

La notion de régions câblées entre elles pour former une hiérarchie nous vient de la biologie. Le cortex cérébral est une grande feuille de tissus composés de neurones de 2mm d'épaisseur que les biologistes divisent en différentes zones ou « régions » principalement en raison des connexions qui les relient entre elles. Certaines régions reçoivent directement leurs informations en entrée depuis les sens alors que pour d'autres elles ont d'abord traversé d'autres régions. C'est la connectivité de région à région qui définit la hiérarchie.

Toutes les régions du cortex cérébral se ressemblent. Elle varie par leur taille et leur emplacement dans la hiérarchie mais, ceci mis à part, elles sont similaires. Si vous pratiquez une coupe dans l'épaisseur de 2mm d'une région du cortex cérébral vous y trouverez 6 couches, 5 couches composées de cellules et une sixième couche non cellulaire (il y a quelques exceptions à ce schéma mais ça représente bien le cas en général). Chaque couche d'une région possède de nombreuses cellules organisées en colonnes. Les régions HTM sont elles aussi composées d'une feuille de cellules fortement interconnectées et disposées en colonnes. La « Couche 3 » du cortex cérébral est l'une des principales couches de neurones à action aval (feed-forward layer). Les cellules d'une région HTM sont à peu près équivalentes aux neurones de la couche 3 d'une région du cortex cérébral.

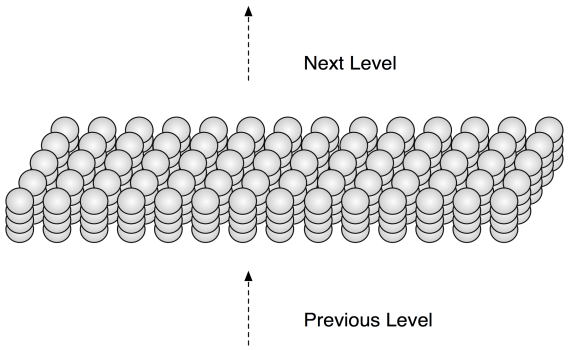


Figure 1.3 : Section d'une région HTM. Les régions HTM se composent de nombreuses cellules. Les cellules sont organisées en un tableau de colonnes à 2 dimensions. Cette figure montre une petite section d'une région HTM avec quatre cellules par colonnes. Chaque colonne est connectée à un sous-ensemble des informations en entrée et chaque cellule est connectée à d'autres cellules dans la même région (ces connexions ne sont pas montrées ici). Il faut noter que cette région HTM, y compris son organisation en colonne, est l'équivalent d'une couche de neurones d'une région du cortex cérébral.

Bien qu'une région HTM n'équivalle qu'à une partie d'une région du cortex cérébral, elle est capable d'inférences et de prédictions sur des flux de données complexes et peut donc être utiles dans la résolution de nombreux problèmes.

Représentations distribuées parcimonieuses

Bien que les neurones du cortex cérébral soient fortement interconnectés, il existe aussi des neurones inhibiteurs qui garantissent qu'à un moment donné seul un

faible pourcentage des neurones est actif. Ainsi, l'information dans le cerveau est toujours représentée par un petit pourcentage de neurones actifs parmi une très grande population de neurones. Ce type d'encodage est appelé « représentation distribuée parcimonieuse » (sparse distributed representation ou SDR). « Parcimonieuse » (sparse) signifie qu'un faible nombre de neurones est actif à un moment donné. « Distribuée » signifie que l'activation simultanée de plusieurs neurones est nécessaire à la représentation de l'information. Un neurone seul est porteur de sens mais il doit être interprété dans le contexte d'une population d'autres neurones actifs pour exprimer tout le sens d'une information.

Les régions MTHs utilisent aussi des représentations distribuées parcimonieuses. En fait, les mécanismes de mémorisation d'une région HTM dépendent directement de cette représentation distribuée parcimonieuse et ne fonctionnerait pas sans elle. L'information d'entrée d'une région HTM est presque toujours distribuée mais elle peut ne pas être parcimonieuse et la première chose que doit donc faire une région HTM est de convertir ses entrées en une représentation distribuée et parcimonieuse.

Prenons à titre d'exemple, une région qui reçoit 20 000 bits d'information en entrée. Le pourcentage de bits d'entrée qui sont à « 0 » ou à « 1 » peut varier de façon significative au fil du temps. À un certain moment il pourrait y avoir 5000 bits à « 1 » et à un autre 9000. La région HTM pourrait convertir ces entrées en une représentation interne de 10 000 bits dont 2% (soit 200) sont actifs simultanément quel que soit le nombre de bits à « 1 » en entrée. Comme les entrées de la région HTM varient avec le temps, la représentation interne évoluera aussi mais il n'y aura toujours que 200 bits actifs sur 10 000.

On pourrait penser que ce processus engendre une grande perte d'information puisque le nombre de motifs possibles en entrée est beaucoup plus grand que le nombre de représentations possibles sur cette région. Toutefois, ces nombres sont tous les deux incroyablement grands. Les véritables entrées vues par une région constitueront une minuscule fraction de toutes les entrées possibles. Plus loin nous décrirons comment une région crée une représentation parcimonieuse de ses entrées. La perte théorique d'information qui en découle n'aura en pratique aucun effet.

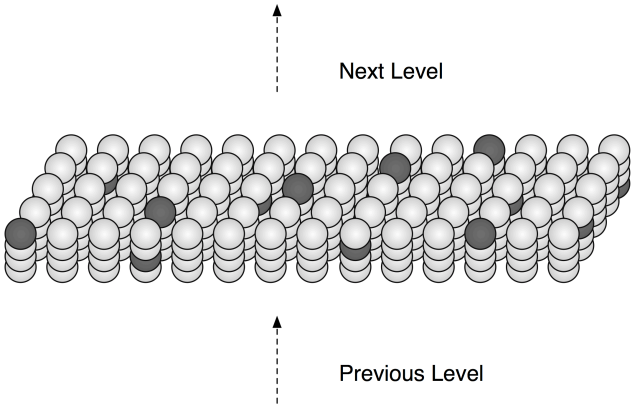


Figure 1.4 : Une région montrant l'activation distribuée et parcimonieuse de cellules

Les représentations distribuées parcimonieuses ont plusieurs propriétés intéressantes et font partie intégrante de la mise en œuvre des MTHs. Nous y reviendrons.

Le rôle du temps

Le temps joue un rôle primordial dans les processus d'apprentissage, d'inférence et de prédiction.

Commençons par l'inférence. Sans la dimension temporelle nous sommes quasiment incapables d'inférer quoique ce soit de notre sens du toucher ou de l'ouïe. C'est ainsi que si les yeux bandés quelqu'un place une pomme dans votre main, vous pouvez l'identifier après l'avoir manipulé après quelques secondes. En déplaçant vos doigts sur la pomme et bien que l'information tactile ne cesse de changer, l'objet lui-même – la pomme ainsi que votre percept « pomme » de haut niveau reste le même. À contrario, si une pomme est placée dans votre paume de main ouverte et qu'il vous est interdit de bouger votre main ou vos doigts vous serez bien en peine de savoir s'il s'agit d'une pomme ou d'un citron.

Il en va de même pour l'ouïe. Un son constant n'a que peu de sens. Un mot comme « pomme » ou le bruit d'une personne mordant une pomme peut seulement être reconnu grâce à la succession de dizaines ou centaines de changements rapides et séquentiels de spectre sonore.

La vue, par contre, est un mélange des deux. Contrairement au toucher et à l'ouïe, les êtres humains sont capables de reconnaitre des images même quand elles leurs sont présentées trop vite pour que l'œil n'ait le temps de bouger. L'inférence visuelle ne nécessite donc pas toujours des entrées qui évoluent dans le temps. Dans la vision normale toutefois nous bougeons constamment les yeux, la tête, le corps et tous les objets autour de nous bougent aussi. Notre capacité d'inférence à partir de temps d'exposition visuelle rapides est un cas particulier rendu possible par les propriétés

statistiques de notre vision et des années d'entrainement. Le cas général pour la vision ainsi que le toucher et l'ouïe est que notre capacité d'inférence nécessite une évolution temporelle des signaux d'entrées.

Après avoir considéré le cas général de l'interface ainsi que le cas particulier de l'inférence visuelle à partir d'images statiques, voyons maintenant l'apprentissage. Pour apprendre, tous les systèmes HTM doivent être exposés durant leur phase d'apprentissage à des informations changeantes dans le temps. Même pour la vision où l'inférence statique est parfois possible, nous devons observer des images d'objets changeantes pour apprendre à quoi ressemble un objet. Imaginez par exemple un chien courant vers vous. À chaque moment le chien engendre un motif sur la rétine de votre œil. Vous percevez ces motifs comme des vues différentes du même chien mais mathématiquement les motifs sont totalement différents. Le cerveau apprend que ces motifs différents représentent la même chose en les observant en séquence. Le temps est le « superviseur » qui vous apprend quels motifs spatiaux vont ensemble.

À noter que l'évolution dans le temps des données sensorielles n'est pas suffisant en soi. Une succession de motifs sensoriels sans lien entre eux n'apporterait que de la confusion. L'évolution temporelle des informations doit émaner d'une source commune. Et remarquons aussi que bien que nous utilisions ici les sens comme exemple, c'est en fait vrai en général. Si nous voulons par exemple entrainer une HTM à reconnaitre des motifs dans l'évolution des températures, des vibrations et des bruits d'une centrale électrique, la HTM devra être précisément entrainée sur les données émanant des capteurs correspondants au fil du temps.

Une HTM doit typiquement être entrainée sur de grandes quantités de données. Vous avez appris à reconnaitre des chiens en voyant de nombreuses instances de multiples races de chiens et pas en observant un seul chien. Le travail des algorithmes HTM consiste à apprendre les séquences temporelles en provenance d'un flux de données c'est-à-dire à construire un modèle de la succession des motifs dans ce flux. C'est une tâche difficile parce qu'il n'est pas toujours possible de savoir où se situent le début et la fin d'une séquence ou bien parce certaines séquences peuvent se recouvrir. L'apprentissage doit donc se dérouler en temps continu et en présence de bruit pour refléter la réalité.

L'apprentissage et la reconnaissance de séquences constituent la base de toute forme de prédiction. Une fois qu'une HTM a appris quels motifs ont des chances d'en suivre d'autres, elle peut prédire quel(s) motif(s) va probablement apparaitre étant donné le flux de données courant et passé. Nous reviendrons en détail sur la prédiction plus loin dans ce document.

Nous allons maintenant nous tourner vers les quatre fonctions fondamentales des HTM : l'apprentissage, l'inférence, la prédiction et le comportement. Toutes les régions HTM effectuent les 3 premières : l'apprentissage, l'inférence et la prédiction. Le comportement, par contre, est différent. Nous savons des biologistes que la

plupart des régions du cortex cérébral jour un rôle dans l'élaboration du comportement mais nous pensons qu'il existe de nombreuses tâches pour lesquels cet aspect n'est pas essentiel. C'est pour cette raison que l'aspect comportement ne figure pas dans notre implémentation actuelle des HTM. Nous le mentionnons ici dans un souci d'exhaustivité.

L'apprentissage

Une région HTM apprend sur le monde qui l'entoure en identifiant des motifs puis des séquences de motifs dans des données sensorielles. La région ne sait pas ce que représentent ces données ; elle travaille dans un monde purement statistique. Elle recherche dans ses entrées des combinaisons de bits qui apparaissent souvent conjointement et que nous appellerons motifs spatiaux. Elle détermine ensuite la façon dont ces motifs spatiaux s'enchainent dans le temps. C'est ce que nous appellerons des motifs temporels ou séquences.

Si les entrées que reçoit une région émanent par exemple des capteurs d'environnement d'un bâtiment, la région pourrait découvrir que certaines combinaisons de température et d'humidité apparaissent souvent sur la façade Nord et d'autres sur la façade Sud. Elle serait aussi capable de déterminer que l'apparition de ces combinaisons se renouvelle quotidiennement.

Si les entrées sont maintenant constituées des ventes d'un magasin, la région HTM pourrait mettre en évidence que certains types d'articles sont achetés le week-end ou que lorsque le temps est froid certaines gammes de prix ont plus de succès en soirée. Ensuite la région pourrait aussi apprendre que des personnes différentes suivent les mêmes motifs séquentiels dans leurs achats.

Une région HTM a des capacités d'apprentissage limitée et elle ajuste automatiquement ce qu'elle apprend en fonction de sa capacité mémoire et de la complexité des entrées qu'elle reçoit. Les motifs spatiaux appris par une région deviendront forcément plus simples si la mémoire allouée décroit et plus sophistiqués si la mémoire augmente. Si les motifs spatiaux appris dans une région sont simples, il sera alors nécessaire de mettre en œuvre une hiérarchie de régions pour apprendre des motifs plus avancés comme ceux qu'on peut trouver dans des images complexes. C'est ce qu'on constate dans le système de vision humain où la région du cortex cérébral qui reçoit les signaux d'entrées en provenance de la rétine apprend des motifs spatiaux correspondants à de toute petite partie de notre champ visuel. Ce n'est qu'après plusieurs niveaux de hiérarchie que les motifs spatiaux sont combinés pour représenter la quasi-totalité du champ visuel.

Comme dans un système biologique, les algorithmes d'apprentissage dans une région HTM sont capables d'apprendre à la volée c'est-à-dire qu'ils continuent à apprendre de chaque nouvelle entrée. Il n'est pas besoin de mettre en place une phase d'apprentissage distincte de la phase d'inférence : la capacité d'inférence de la

région s'améliore avec chaque nouvelle entrée apprise. Ainsi si les motifs reçus évoluent, la région changera elle aussi graduellement.

Après une phase d'apprentissage initial, une région HTM peut soit continuer à apprendre soit suspendre l'apprentissage. Il est aussi possible de suspendre la fonction d'apprentissage dans les niveaux les plus bas d'une hiérarchie mais continuer à apprendre dans les niveaux hauts. Une fois qu'une région a acquis les structures statistiques de base du monde qui l'entoure la majorité des apprentissages se déroulent dans les niveaux supérieurs de la hiérarchie. Si une région HTM est exposée à de nouveaux motifs utilisant des structures de bas niveau jamais vu jusque-là, l'apprentissage prendra plus de temps. C'est un phénomène bien connu chez les êtres humains. Apprendre des mots d'une langue étrangère composés de sons inhabituels est beaucoup plus difficile précisément parce que les sons de base ne sont pas connus.

Le simple fait de pouvoir découvrir des motifs est une capacité potentiellement précieuse. Comprendre les motifs de haut niveau dans les fluctuations des marchés, des maladies, du temps, des rendements de fabrication ou des pannes de systèmes complexes tels que des réseaux électriques est précieux en soi. Malgré cela, l'apprentissage de motifs spatiaux et temporels n'est qu'un préalable à l'inférence et à la prédiction.

Inférence

Après la phase d'apprentissage une HTM est en mesure de produire des inférences sur de nouvelles entrées. Lorsqu'une HTM reçoit une entrée, elle la met en correspondance avec des motifs spatiaux et temporels appris précédemment. Mettre avec succès de nouvelles entrées en correspondance avec des séquences déjà connues est l'essence même de l'inférence et de la reconnaissance de motifs.

Songez à la façon dont vous reconnaissez une mélodie. L'écoute de la première note ne vous renseigne que très peu. La seconde réduit considérablement l'ensemble des possibilités mais cela peut ne pas suffire. En général il faut trois, quatre notes ou plus avant que vous ne soyez en mesure de reconnaitre une mélodie. Le processus d'inférence d'une région HTM procède de façon similaire. La région scrute constamment le flux d'information en entrée à la recherche de ces précieux motifs appris précédemment. Une région HTM est capable d'identifier des correspondances en partant du début d'un flux d'information mais elle est aussi beaucoup plus flexible, comme vous pouvez l'être vous-même en vous montrant capable de reconnaitre une mélodie à n'importe quel moment de l'écoute et pas uniquement en partant de la première note. Les régions HTM s'appuyant sur des représentations distribuées, l'utilisation par la région de la mémorisation et de l'inférence de séquences est plus sophistiquée que l'exemple de la mélodie ne le laisse penser mais ça vous donne idée de son fonctionnement.

Ce n'est peut-être pas immédiatement évident mais bien que chacune de vos expériences sensorielles soit unique vous êtes toujours en mesure d'y retrouver des motifs familiers. Par exemple vous pouvez comprendre le mot « déjeuner » prononcé par à peu près n'importe quelle personne qu'elle soit jeune ou âgée, homme ou femme, parlant vite ou lentement ou ayant un fort accent. Et même si vous demandiez à la même personne de répéter cent fois le mot « déjeuner », le son produit ne solliciterait jamais exactement votre cochlée (votre récepteur auditif) de la même façon deux fois de suite.

Une région HTM est confrontée au même problème que votre cerveau : les informations fournies en entrée ne se répètent jamais exactement de la même façon. Aussi, comme votre cerveau, une région HTM doit traiter une nouvelle entrée aussi bien en inférence qu'en apprentissage. L'utilisation de représentations distribuées parcimonieuses est un des moyens qui permet à une région HTM de faire face à une nouvelle entrée. Une des propriétés clés de ces représentations fait que vous n'avez besoin de mettre en correspondance qu'une portion d'un motif pour être confiant sur son niveau de pertinence.

Prédiction

Chaque région d'une HTM stocke des séquences de motifs. En appariant des séquences mémorisées avec l'état courant d'une entrée, une région forme des prédictions sur les états probables à venir. Les régions HTM mémorisent en fait des transitions entre représentations distribuées parcimonieuses. Dans certains cas les transitions se présentent comme des séquences linéaires comme les notes d'une mélodie mais en général la HTM prédit simultanément de nombreuses variantes de l'état futur d'une entrée, prédictions fondées sur un contexte qui peut remonter loin dans le temps. La plus grande partie de la mémoire d'une HTM est dédiée à la mémorisation de séquences ou au stockage de transitions entre motifs spatiaux.

Voici quelques-unes des propriétés clés du processus de prédiction d'une HTM :

1) La prédiction est continue
Sans en avoir conscience, nous sommes constamment en mode prédiction. Les MTHs font la même chose. Lorsque vous écoutez une chanson, vous êtes tout le temps en train de prédire la prochaine note. Lorsque vous descendez un escalier, vous prédisez le moment où votre pied va rencontrer la prochaine marche. Lorsque vous regardez un lancer de balle au baseball, vous prédisez son arrivée sur le batteur. La prédiction est partie intégrante du fonctionnement d'une HTM.

2) La prédiction se produit dans toutes les régions à tous les niveaux de la hiérarchie
Si vous avez une hiérarchie de régions HTM la prédiction se déroule à tous les niveaux. Chaque région fera des prédictions sur la base des motifs qu'elle a déjà

appris. Si nous prenons l'exemple du langage, les régions de plus bas niveaux prédiront les prochains phonèmes possibles et les régions de plus haut niveau prédiront les prochains mots ou phrases.

3) Les prédictions dépendent du contexte

Les prédictions se font sur la base de ce qui est arrivé dans le passé et de ce qui est en train de se passer. Ainsi une entrée donnée produira des prédictions différentes sur la base du contexte antérieur. Une région HTM apprend à utiliser autant de contextes passés que nécessaire et peut conserver ces contextes à la fois sur des périodes de temps courtes et longues. Cette capacité est connue sous le nom de mémoire d'ordre variable. Songez par exemple à un discours célèbre que vous auriez mémorisé comme le discours d'Abraham Lincoln à Gettysburg[2]. Prédire le prochain mot du discours si on vous en donne juste un pris au hasard est rarement suffisant. En effet ce mot peut-être présent plusieurs fois dans le discours dans des contextes différents. Parfois avec un peu plus de contexte tel que 2 mots d'affilés votre cerveau prédira la suite de la phrase. De la même façon votre cerveau est capable d'accumuler un contexte beaucoup plus long de 3, 4, 5 mots ou plus avant de déterminer à coup sur le prochain mot.

4) La prédiction amène de la stabilité.

La sortie d'une région est sa prédiction. L'une des propriétés de MTHs tient au fait que les sorties produites par régions deviennent plus stable – c'est-à-dire évoluant plus lentement et durant plus longtemps – au fur et à mesure qu'on monte dans la hiérarchie. Cette propriété résulte directement de la façon dont une région établit sa prédiction. En effet une région ne se contentera pas de prédire ce qui va immédiatement survenir mais ce qui peut se passer sur plusieurs étapes de temps à venir. Supposons qu'une région puisse prédire jusqu'à cinq étapes en avance. Lorsqu'une nouvelle entrée arrive, la nouvelle étape prédite change mais les quatre autres déjà établie peuvent ne pas changer. Ainsi, bien que chaque étape soit complètement différente, seule un partie de la sortie change, rendant les sorties plus stable que les entrées. Cette caractéristique reflète notre expérience du monde réel où les concepts de haut niveau – comme le titre d'une chanson – changent beaucoup plus lentement que les concepts de bas niveau – comme les notes de cette chanson.

5) Une prédiction nous indique si une nouvelle entrée était attendue ou pas.

Chaque région HTM est un détecteur de nouveautés. Puisque qu'une région prédit en permanence ce qui va produire, elle est aussi capable de signaler l'apparition de quelque chose d'inattendu. Les MTHs peuvent prédire plusieurs prochaines entrées de façon simultanée ; pas uniquement une seule. Elle peut donc ne pas prédire exactement ce qui va arriver ensuite, mais elle sera toujours capable de signaler qu'une entrée ne correspond à aucune des prédictions de la région HTM et qu'une anomalie vient de survenir.

[2] Note du traducteur : pour les lecteurs peu familiers de l'histoire des Etats Unis d'Amérique voir https://en.wikipedia.org/wiki/Gettysburg_Address

6) La prédiction rend le système plus résistant au bruit.
Quand une HTM prédit ce qui va très probablement survenir, la prédiction a pour effet d'orienter le système vers ces inférences. Par exemple, une HTM qui traite une langue parlée va prédire quels sons, quels mots et quelles idées vont surement être émis par le locuteur. Cette prédiction aide en fait le système à boucher les trous. Si un son ambigu survient, la HTM l'interprétera en fonction de ce qu'elle attend facilitant ainsi le processus d'inférence même en présence de bruit.

Dans une région HTM, la mémoire des séquences, l'inférence et la prédiction sont intimement liées. Elles constituent les fonctions centrales d'une région.

Comportement

Notre comportement influence ce que nous percevons. Quand nos yeux bougent notre rétine reçoit des informations changeantes. Les mouvements de nos membres et de nos doigts envoient des sensations de toucher toujours différentes vers notre cerveau. La quasi-totalité de nos actions influent sur ce que nous ressentons. La perception sensitive et notre comportement moteur sont intimement liés.

Pendant des décennies l'opinion générale était qu'une région bien spécifique du cortex cérébral, la région motrice primaire, était le siège des commandes motrices. Au fil du temps on a découvert que quasiment toutes les régions du cortex ont une sortie motrice y compris les régions sensorielles de bas niveau. Il apparait aujourd'hui que toutes les régions du cortex intègrent à la fois les fonctions sensorielles et motrices.

Nous pensons qu'une sortie motrice pourrait être ajoutée à chaque région HTM dans le cadre actuel puisque générer des commandes motrices s'apparente à l'élaboration de prédictions. Toutefois, jusqu'à présent toutes les implémentations des MTHs sont purement sensorielles et ne comportent aucun composant moteur.

Avancées vers une implémentation de la HTM

À partir du cadre théorique de la HTM nous avons fait de réels progrès pour en tirer une technologie applicable. Nous avons implémenté et testé plusieurs versions des algorithmes d'apprentissage cortical HTM et nous avons pu constater que l'architecture de base est solide. Au fur et à mesure de nos tests sur de nouveaux jeux de données, nous améliorerons les algorithmes et ajouterons les éléments manquants. Ce document sera mis à jour en conséquence. Les trois chapitres qui suivent décrivent l'état actuel des algorithmes.

De nombreux aspects de la théorie ne sont pas encore implémentés y compris l'attention, le feedback entre régions, le cadencement temporel précis et

l'intégration comportement/sensori-motricité. Tous ces aspects devraient toutefois trouver leur place dans le cadre théorique déjà créé.

Chapitre 2: les algorithmes d'apprentissage corticaux HTM

Ce chapitre décrit les algorithmes d'apprentissage en action dans une région HTM. Les chapitres 3 et 4 décrivent leur implémentation en pseudocode alors que celui-ci se focalise davantage sur les concepts.

Terminologie

Avant de commencer, un peu de terminologie peut s'avérer utile. Pour décrire les algorithmes d'apprentissage HTM nous utilisons le langage des neurosciences. Des termes tels que cellules, synapses, synapses potentielles, segments dendritiques et colonnes sont utilisés en permanence. Cette terminologie parait logique puisque les algorithmes d'apprentissage sont très largement dérivés d'une mise en correspondance des connaissances détaillées des neurosciences avec nos besoins sur le plan théorique. Cependant, lors de l'implémentation des algorithmes nous avons été confrontés à des problèmes de performance et par conséquent, une fois que nous pensions avoir compris leur fonctionnement, nous avons donc cherché à accélérer les temps de traitement. Pour ce faire nous avons souvent dévié de la stricte conformité aux détails biologiques pour autant que les résultats finaux restaient inchangés. Si vous êtes un nouveau venu dans les neurosciences ce ne sera pas un problème. Dans le cas contraire vous éprouverez peut-être une certaine confusion car notre utilisation des termes des neurosciences ne correspondront pas à vos habitudes. L'annexe sur les aspects biologiques couvre en détail les différences et les similarités entre les algorithmes d'apprentissage HTM et leurs équivalents neurobiologiques. Nous mentionnons ici quelques-unes des déviations qui causeront très probablement une certaine confusion.

Les états d'une cellule
Les cellules HTM possèdent trois états de sortie : activée par une entrée aval (feed-forward), activée par une entrée latérale et désactivée. Le premier état correspond à une courte rafale de potentiels d'action d'un neurone. Le second à une rafale plus lente et régulière des potentiels d'action. Nous n'avons pas ressenti le besoin de modéliser des potentiels d'action individuels ni même de quantifier le taux d'activité par une grandeur scalaire au-delà de ces deux états actifs. L'utilisation des représentations distribuées semble éliminer le besoin de modéliser par une grandeur scalaire les taux d'activité dans les cellules.

Les segments de dendrites
Les cellules HTM utilisent un modèle de dendrites relativement proche de la réalité (et donc complexe). En théorie chaque cellule HTM possède un segment de dendrites proximales et une ou deux douzaines de segments de dendrites distales. Le segment de dendrites proximales reçoit l'entrée aval et le segment de dendrites distales reçoit l'entrée latérale des cellules proches. Une classe de cellules inhibitrices oblige toutes les cellules d'une colonne à répondre à la même entrée

aval. Pour simplifier, nous avons remplacé le segment de dendrites proximales de chacune des cellules par un seul segment de dendrites partagé commun à toutes les cellules. La fonction de répartition spatiale (décrite plus bas) opère sur le segment de dendrites partagé au niveau des colonnes. La fonction de répartition temporelle opère sur les segments de dendrites distales au niveau de chacune des cellules dans les colonnes. Bien qu'en biologie il n'existe aucun équivalent à ce segment de dendrites attaché à la colonne, cette simplification aboutit cependant à un niveau de fonctionnalité identique.

Synapses

Les synapses HTM ont des poids binaires. Les synapses biologiques utilisent certes des poids variables mais qui sont aussi partiellement stochastiques ce qui laisse penser qu'un neurone biologique ne peut pas baser son fonctionnement sur des poids synaptiques précis. L'utilisation de représentations distribuées dans les MTHs ajoutée au modèle opératoire des dendrites nous permet d'affecter des poids binaires aux synapses des MTHs sans effet néfaste. Pour modéliser la formation et la disparition de synapses nous utilisons deux concepts supplémentaires issus des neurosciences avec lesquels vous n'êtes peut-être pas familier. Le premier est le concept de « synapses potentielles ». Il représente tous les axones qui passent suffisamment prêt d'un segment de dendrites pour pouvoir potentiellement former une synapse. Le second est appelé « permanence » et représente une valeur scalaire assignée à chaque synapse potentielle. La permanence d'une synapse représente un niveau de connexité entre un axone et une dendrite allant typiquement d'une synapse totalement déconnectée, à une synapse en formation mais pas encore connectée, puis connectée de façon minimale pour finir sur une synapse totalement connectée. La permanence est une valeur scalaire entre 0.0 et 1.0. La phase d'apprentissage se charge d'incrémenter ou de décrémenter la permanence d'une synapse. Lorsqu'un certain seuil est franchi elle devient totalement connectée avec un poids de « 1 ». En dessous de ce seuil elle est déconnectée et son poids est de « 0 ».

Vue d'ensemble

Imaginez que vous soyez une région d'une HTM. Vos entrées sont constituées de milliers ou dizaines de milliers de bits. Ces bits d'entrée peuvent représenter des données sensorielles ou provenir d'une autre région située plus bas dans la hiérarchie et changent d'état de manière complexe. Qu'êtes-vous donc supposé faire avec ces entrées ?

Nous avons déjà répondu à cette question dans sa forme la plus simple. Chaque région HTM recherche des motifs communs dans ces entrées et apprend ensuite les séquences dans lesquelles ces motifs s'enchainent. À partir de cette mémoire des séquences chaque région élabore des prédictions. Cette description de haut niveau

fait paraitre la tâche très simple mais en réalité il se passe beaucoup de choses dans ce processus. Découpons-le un peu plus finement en 3 étapes :

1) Former une SDR des entrées
2) Former une représentation des entrées en tenant du contexte des entrées antérieures
3) Former une prédiction basée sur les entrées courantes dans le contexte des entrées antérieures

Nous allons évoquer chacune de ces étapes plus en détails.

1) Former une SDR des entrées

Lorsque vous imaginez les entrées soumises à une région, voyez-la comme un très grand nombre de bits. Dans un cerveau cela correspondrait aux axones des neurones. À un moment donné certaines de ces entrées sont actives (valeur 1) et d'autres inactives (valeur 0). Le pourcentage de bits d'entrée actifs varie dans le temps disons entre 0% et 60%. La première chose qu'une région HTM va faire consiste à convertir ces entrées en une nouvelle représentation dite parcimonieuse (c'est-à-dire plus économe en quantité d'information). Ainsi une combinaison d'entrées avec 40% de bits actifs se transformera en une nouvelle représentation où seuls 2% des bits le sont.

D'un point de vue logique, une région HTM est faite d'un ensemble de colonnes. Chaque colonne est composée d'une ou plusieurs cellules. Les colonnes peuvent être organisées selon un tableau à deux dimensions mais ce n'est pas une obligation. Chaque colonne est connectée à un sous-ensemble unique des bits d'entrées (habituellement en recouvrement avec les entrées d'autres colonnes mais jamais exactement le même). De cette façon, des motifs d'entrée différents donnent différents niveau d'activation dans les colonnes. Les colonnes avec le plus fort niveau d'activation inhibent ou désactivent les colonnes les plus faiblement activées (l'inhibition survient dans un rayon qui peut être très local ou bien sur la région entière). La représentation parcimonieuse des entrées est encodée par le biais de colonnes actives ou inactives après cette phase d'inhibition. La fonction d'inhibition est définie de façon à atteindre un pourcentage relativement constant de colonnes actives même lorsque le nombre de bits d'entrée actifs varie considérablement.

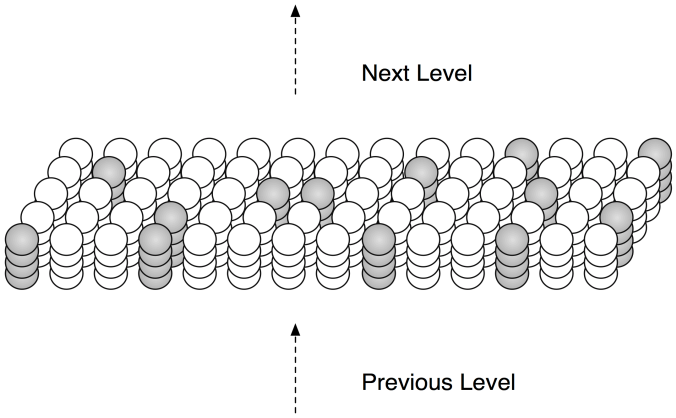


Figure 2.1: Une région HTM est faite de colonnes de cellules Seule une petite partie de la région est montrée ici. Chaque colonne de cellules reçoit son ordre d'activation d'un sous ensemble unique de bits d'entrée. Les colonnes avec la plus forte activation inhibent les colonnes plus faibles. Le résultat est une SDR des entrées. La figure montre les colonnes actives en gris clair. (Lorsqu'une colonne n'avait pas d'état antérieur, toutes les cellules de la colonne active sont elles-mêmes actives comme représenté ici).

Imaginez maintenant que le motif d'entrée change. Si seuls quelques bits d'entrée sont modifiés, certaines colonnes recevront des entrées actives en plus ou en moins mais l'ensemble des colonnes actives ne changera probablement pas beaucoup. Ainsi des motifs similaires (c'est-à-dire présentant un nombre important de bits actifs communs) seront mis en correspondance avec un ensemble de colonnes relativement stable. Le degré de stabilité dépend largement des entrées auxquelles chaque colonne est reliée. Ces connexions résultent d'un apprentissage que nous décrirons plus loin.

Toutes ces étapes (apprendre les connexions à chaque colonne à partir d'un sous-ensemble d'entrées, déterminer le niveau d'entrée de chaque colonne et utiliser le processus d'inhibition pour sélectionner un ensemble parcimonieux de colonnes actives) sont regroupées sous la dénomination « Concentrateur spatial ». Le terme signifie que les motifs qui sont « spatialement » similaires (c'est-à-dire ayant en commun un grand nombre de bits actifs) sont « concentrés » (c'est-à-dire regroupés) dans une représentation commune.

2) Former une représentation des entrées en tenant compte du contexte des entrées antérieures

L'autre fonction remplie par une région consiste à convertir la représentation en colonne des entrées en une nouvelle représentation interne qui prend en compte l'état (ou contexte) passé. La nouvelle représentation est construite en activant un sous-ensemble des cellules dans chaque colonne (typiquement une seule par colonne – Voir Figure 2.2).

Supposez que vous entendiez les 2 phrases « Je scie la chaise » et « J'ai six chaises ». Les mots « scie » et « six » sont des homonymes ; ils sonnent de la même façon. Nous

pouvons être certains qu'à un moment dans le cerveau il existe des neurones qui vont répondre à l'identique aux deux mots puisqu'après tout, les sons qui entrent dans l'oreille sont semblables. Mais nous pouvons aussi être certains qu'à un autre endroit dans le cerveau il existe des neurones qui vont répondre différemment aux deux mots compte tenu des contextes qui eux sont bien différents. Les représentations pour le son « scie » seront différentes lorsque vous entendez « Je scie » d'un côté et « J'ai six » de l'autre. En imaginant que vous ayez déjà mémorisé par le passé les 2 phrases « Je scie la chaise » et « J'ai six chaises », le fait d'entendre « Je scie… » déclenchera une prédiction différente de celle déclenchée par « J'ai six… ». On doit donc avoir 2 représentations internes différentes après avoir entendu « Je scie… » et « J'ai six… ».

Ce principe qui consiste à encoder différemment des entrées dans des contextes distincts est une caractéristique universelle des fonctions de perception et d'action et c'est l'une des fonctions les plus importantes d'une région HTM. On n'insistera jamais assez sur l'importance de cette propriété des MTHs.

Chaque colonne dans une région HTM est constituée de multiples cellules. Toutes les cellules d'une colonne reçoivent la même entrée aval. Chaque cellule peut être active ou pas. En sélectionnant des cellules actives différentes dans une colonne active, il est possible de représenter la même entrée de façon différente en fonction du contexte. Prenons un exemple pour faciliter la compréhension. Supposons que chaque colonne ait 4 cellules et que la représentation des entrées se fasse sur 100 colonnes actives. Si une seule cellule par colonne est active à tout moment nous disposons de 4^100 façons de représenter exactement la même entrée. En d'autres termes, la même entrée causera toujours l'activation des mêmes 100 colonnes mais des contextes différents pourront activer des cellules différentes dans ces colonnes. Nous pouvons donc représenter la même entrée dans un très grand nombre de contextes mais à quel point chacune de ces représentations sera-t-elle unique ? À peu près toutes les paires possibles des 4^100 motifs possibles auront environ 25 cellules en commun. Ainsi deux représentations d'une entrée donnée dans différents contextes auront environ 25 cellules en commun et 75 cellules différentes ce qui permet de les distinguer facilement.

La règle générale utilisée par une région HTM est la suivante : lorsqu'une colonne devient active elle regarde toutes ses cellules. Si une ou plusieurs cellules dans la colonne sont déjà en état prédictif seules ces cellules deviennent actives. Si aucune cellule n'est en état prédictif alors toutes les cellules deviennent actives. Vous pouvez aussi le formuler ainsi : si un motif d'entrée est attendu alors le système indique cette attente en activant seulement les cellules en état prédictif. Si le motif d'entrée est inattendu alors le système active toutes les cellules de la colonne comme pour dire « cette entrée survient alors que je ne m'y attendais pas donc toutes les interprétations restent valides ».

S'il n'existe aucun état antérieur, et donc aucun contexte ni aucune prédiction, toutes les cellules d'une colonne deviendront actives lorsque la colonne devient

active. Ce scénario s'apparente à ce qui se passe lorsque vous écoutez la première note d'une chanson. Sans contexte vous ne pourrez pas prédire ce qui va se passer ensuite ; toutes les options sont ouvertes. S'il existe un état antérieur mais que l'entrée ne correspond pas à ce qui est attendu, toutes les cellules de la colonne deviendront actives. Ce processus s'applique colonne par colonne ce qui signifie que la prédiction d'une correspondance ou bien d'une non correspondance n'est jamais un évènement de type tout ou rien.

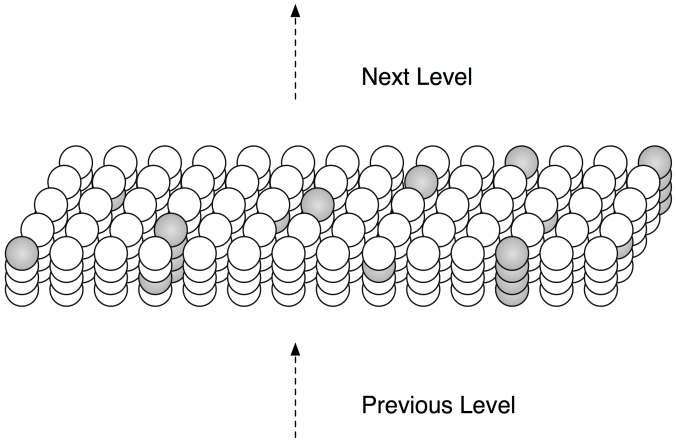


Figure 2.2: En activant un sous-ensemble des cellules de chaque colonne, une région HTM peut représenter une même entrée dans de nombreux contextes tous différents. Les colonnes activent uniquement les cellules prédites. Les colonnes sans cellules prédites activent toutes les cellules dans la colonne. La figure montre certaines colonnes avec une seule cellule activée et d'autres avec toutes les cellules activées.

.

Comme indiqué dans la section ci-dessus consacrée à la terminologie, les cellules d'une HTM peuvent se trouver dans un seul état parmi trois. Si une cellule est active en raison d'une entrée aval (feed-forward input) on utilise simplement le terme « active ». Si la cellule est active en raison d'une connexion latérale à d'autres cellules proches on dit qu'elle est dans l' « état prédictif » (Figure 2.3).

3) Former une représentation des entrées en tenant du contexte des entrées antérieures

La dernière étape pour notre région consiste à prédire ce qui va probablement se passer juste après. La prédiction s'appuie sur la représentation construite à l'étape 2) qui comprend le contexte de toutes les entrées précédentes.

Lorsqu'une région forme une prédiction elle active (dans l'état prédictif) toutes les cellules qui vont très probablement devenir actives en raison de la prochaine entrée aval. Etant donné que les représentations dans une région sont parcimonieuses, plusieurs prédictions peuvent être faites simultanément. Par exemple si 2% des colonnes sont actives en raison d'une entrée, il se pourrait que 10 prédictions différentes soient faites amenant à 20% des colonnes avec une cellule en état prédictif. Ou bien 20 prédictions différentes pourraient être faites amenant à 40%

des colonnes avec une cellule en état prédictif. Si chaque colonne possède 4 cellules, avec une seule d'entre elles active à un moment donné, alors 10% des cellules seraient dans un état prédictif.

Un prochain chapitre consacré aux représentations distribuées parcimonieuses montrera que même si des prédictions différentes sont fusionnées entre elles, une région a le moyen de savoir avec un haut degré de certitude si une entrée particulière a été prédite ou pas.

Comment une région fait-elle pour former une prédiction ? Lorsque les motifs en entrée changent au fil du temps, des sous-ensembles différents de colonnes et de cellules sont successivement activées. Lorsqu'une cellule devient active, elle forme des connexions vers un sous-ensemble de cellules situées à proximité qui étaient déjà actives juste avant. Ces connexions peuvent se former rapidement ou lentement suivant le rythme d'apprentissage imposé par l'application. Plus tard, tout ce qu'une cellule doit faire c'est regarder quelles sont ces connexions ayant une activité coïncidente. Si les connexions deviennent actives, la cellule peut s'attendre à devenir possiblement active assez vite et passer à l'état prédictif. Ainsi l'activation par une entrée aval (feed-forward activation) d'un jeu de cellules amènera à l'activation prédictive d'autres jeux de cellules à sa suite. Vous pouvez vous figurer ce processus comme ce que vous faites lorsqu'après avoir reconnu une chanson vous commencez à prévoir (prédire) les notes à venir.

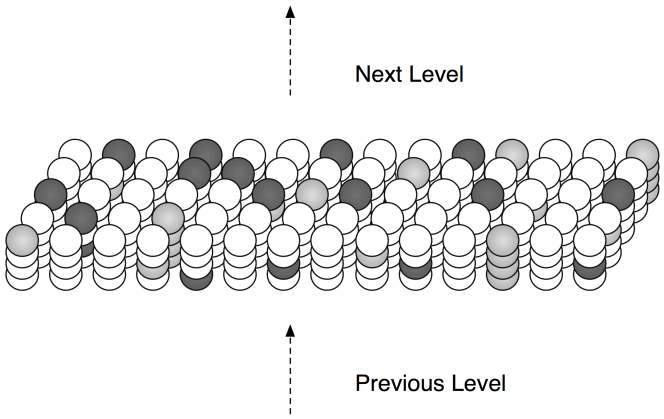


Figure 2.3: À tout moment, certaines cellules d'une région HTM sont activées par une entrée aval (feed-forward input) (montrées ici en gris clair). D'autres cellules qui reçoivent des entrées latérales depuis des cellules actives passent à l'état prédictif (montrées en gris foncé).

En résumé, lorsqu'une entrée arrive elle active un ensemble parcimonieux de colonnes. Une ou plusieurs cellules de chaque colonne deviennent actives, provoquant le passage d'autres cellules à l'état prédictif par le biais de connexions apprises entre cellules de la région. Les cellules activées via ces connexions au sein de la région forme une prédiction ou, en d'autres termes, ce qui a de fortes chances

de survenir ensuite. Lorsque l'entrée aval (feed-forward input) suivante arrive , elle active un autre ensemble parcimonieux de colonnes. Si une colonne nouvellement active n'était pas attendue, autrement si elle n'était prédite par aucune de ses cellules, toutes les cellules de cette colonne passent à l'état actif. Si une colonne nouvellement activée avait déjà une ou plusieurs cellules à l'état prédictif alors elles deviennent actives. La sortie d'une région est constituée de l'activité de toutes les cellules de la région, c'est-à-dire à la fois des cellules actives suite à une entrée aval *et* des cellules actives à l'état prédictif.

Comme indiqué précédemment, les prédictions ne concernent pas uniquement la *prochaine* étape dans le temps. Les prédictions effectuées par une région HTM peuvent concernées des étapes plus éloignées dans le futur. Avec une mélodie par exemple, une région HTM ne produira pas uniquement la prochaine note mais pourrait très bien prédire les 4 prochaines notes. Ce qui nous amène à une propriété tout à fait intéressante : la sortie d'une région (c'est-à-dire l'union de toutes les cellules d'une région en état actif ou prédictif) change moins vite que l'entrée. Imaginez que la région prédise les 4 prochaines notes d'une mélodie composée des notes suivantes en séquence La, Si, Do, Ré, Mi, Fa, Sol. Après avoir entendu les deux premières notes (La, Si), la région reconnait la séquence et commence à faire des prédictions, en l'occurrence Do, Ré, Mi, Fa. Les cellules « Si » sont déjà actives donc les cellules qui codent pour Si, Do, Ré, Mi, Fa sont toutes dans un des deux états actifs. Puis la région entend la prochaine note « Do ». L'ensemble des cellules en état actif ou prédictif représentent maintenant « Do, Ré, Mi, Fa, Sol ». Vous remarquez donc que le motif d'entrée a complètement changé passant de « Si » à « Do » mais que seuls 20% des cellules ont changé d'état (1 note sur 5).

Comme la sortie d'une région est un vecteur qui représente l'activité de toutes les cellules de la région, la sortie dans cet exemple est cinq fois plus stable que l'entrée. Dans une organisation hiérarchique des régions, nous observerons un accroissement de la stabilité temporelle au fur et à mesure que nous montons dans la hiérarchie.

Nous utilisons le terme « concentrateur temporel » pour décrire les deux étapes qui consistent à ajouter un contexte à une représentation et former une prédiction. En créant des sorties qui varient lentement dans le temps à partir des motifs d'entrées très variables, nous « concentrons » (ou regroupons) des motifs différents qui se suivent dans le temps.

Nous allons maintenant passer à un autre niveau de détails. Nous allons commencer par des concepts communs au concentrateur spatial et au concentrateur temporel. Ensuite nous aborderons les concepts spécifiques au concentrateur spatial puis au concentrateur temporel.

Concepts communs

Les processus d'apprentissage dans le concentrateur spatial et dans le concentrateur temporel sont très similaires. Dans les 2 cas l'apprentissage implique la mise en place de connexions, ou synapses, entre cellules. Le concentrateur temporel apprend en mettant en place les connexions entre cellules d'une même région. Le concentrateur spatial apprend en créant les connexions entre les bits d'entrée et les colonnes.

Poids binaires
Les synapses d'une HTM ont un effet de type tout (1) ou rien (0). Leur « poids » est binaire contrairement à de nombreux modèles de réseaux de neurones qui utilisent des valeurs scalaires dans l'intervalle 0 à 1.

Permanence
Les synapses se font et se défont constamment durant la phase d'apprentissage. Comme mentionné précédemment, nous assignons une valeur scalaire à chaque synapse (entre 0.0 et 1.0) pour indiquer le degré de permanence d'une connexion. Lorsqu'une connexion est renforcée sa permanence croit et elle décroit dans d'autres conditions. Lorsque la permanence dépasse un certain seuil (0.2), la synapse (et donc la connexion) est considérée comme établie. Si la permanence passe en dessous du seuil, la synapse n'aura aucun effet.

Segments de dendrites
Les synapses se connectent à des segments de dendrites. Il en existe 2 sortes : proximal et distal.
- Un segment de dendrites proximal forme des synapses avec des entrées aval (feed-forward inputs). Les synapses actives sur ce type de segment sont additionnées de façon linéaire pour déterminer l'activation à partir des entrées aval d'une colonne.
- Un segment de dendrites distales forme des synapses avec d'autres cellules de la région. Chaque cellule a plusieurs segments de dendrites distaux. Si la somme des synapses actives sur un segment distal dépasse un certain seuil, les cellules associées deviennent actives en état prédictif. Comme il y a plusieurs segments de dendrites distaux par cellule, l'état prédictif de la cellule résulte d'un OU logique de tous les détecteurs de dépassement de seuil.

Synapses potentielles
Comme mentionné précédemment, chaque segment de dendrites possède une liste de synapses potentielles. Chaque synapse potentielle possède une valeur de permanence et devient opérationnelle si cette valeur excède un certain seuil.

Apprentissage
Le processus d'apprentissage repose sur l'incrémentation ou la décrémentation des valeurs de permanence des synapses potentielles disposées sur les segments de

dendrites. Les règles utilisées pour rendre les synapses plus ou moins permanentes sont similaires aux règles d'apprentissage de Hebb. Par exemple, si une cellule post-synaptique est active en raison d'un segment de dendrite recevant des entrées qui dépassent son seuil d'activation, alors les valeurs de permanence des synapses sur ce segment sont modifiées. Les synapses qui sont actives, et donc qui ont contribué à rendre la cellule active, voient leur permanence croitre. Les synapses qui sont inactives, et qui n'ont donc pas contribué à l'activation de la cellule, voient leur permanence décroitre. Les circonstances exactes dans lesquelles les valeurs de permanence sont mises à jour diffèrent selon qu'on se trouve dans le concentrateur spatial ou le concentrateur temporel. Les détails sont fournis plus loin.

Nous allons maintenant aborder les concepts spécifiques à chacun des concentrateurs, spatial et temporel.

Concepts du concentrateur spatial

La fonction la plus fondamentale du concentrateur spatial consiste à convertir les entrées d'une région en un motif parcimonieux (c'est-à-dire économe en information). Cette fonction est importante car le mécanisme utilisé pour apprendre des séquences et former des prédictions requiert ces motifs distribués parcimonieux.

Le concentrateur spatial doit remplir plusieurs objectifs qui déterminent son mode opératoire et sa façon d'apprendre.

1) Utiliser toutes les colonnes
Une région HTM possède un nombre fixe de colonnes qui apprennent à représenter les motifs récurrents qui apparaissent sur les entrées. L'un des objectifs est de s'assurer que toutes les colonnes apprennent à représenter quelque chose d'utile indépendamment du nombre de colonnes présentes. Nous devons par exemple éviter d'avoir des colonnes qui ne seraient jamais actives. Pour cela, nous gardons une trace de la fréquence d'activation d'une colonne par rapport à ses voisines. Si l'activité relative d'une colonne est trop faible, elle dope son niveau d'activité en entrée jusqu'à ce qu'elle recommence à faire partie de l'ensemble de colonnes gagnantes. Fondamentalement, toutes les colonnes sont en compétition avec leurs voisines pour participer à la représentation d'une entrée. Si une colonne n'est pas très active, elle deviendra plus agressive. Lorsque c'est le cas, les autres colonnes se verront contraintes de modifier leurs entrées et de commencer à représenter des motifs en entrée légèrement différents.

2) Maintenir la densité désirée
Une région a besoin de former une représentation distribuée de ses entrées. Les colonnes avec le plus d'entrées inhibent leurs voisines. Il existe un rayon d'inhibition qui est proportionnel à la taille du champ de réception des colonnes (et

peut donc être petit ou bien s'étendre à toute la région). Dans le périmètre défini par ce rayon, nous autorisons uniquement un certain pourcentage des colonnes avec les entrées les plus actives à être « gagnantes ». Les autres colonnes sont désactivées. (Un « rayon » d'inhibition implique un arrangement des colonnes en 2D mais le concept peut être transposé à d'autres topologies.)

3) Eviter les motifs triviaux

Nous voulons que toutes nos colonnes représentent des motifs non triviaux sur les entrées. On peut y parvenir en positionnant un seuil minimum d'entrées pour que la colonne devienne active. Ainsi, si le seuil est positionné à 50, cela signifie que la colonne doit avoir au moins 50 synapses actives sur son segment de dendrites pour devenir active, garantissant ainsi un certain niveau de complexité sur le motif représenté.

4) Eviter l'excès de connexions

Si nous n'y prenons garde, une colonne pourrait former un grand nombre de synapses valides et ce faisant répondre à un grand nombre de motifs d'entrée différents et sans aucun lien entre eux. Des sous-ensembles distincts de synapses répondraient alors à des motifs différents. Pour éviter ce problème, nous décrémentons la valeur de permanence de toute synapse qui n'est pas en train de contribuer à une colonne gagnante. En s'assurant que les synapses non contributrices sont suffisamment pénalisées, nous pouvons garantir qu'une colonne représente un nombre limité de motifs d'entrée et parfois un seul.

5) Des champs réceptifs auto-ajustables

Les cerveaux biologiques sont hautement « plastiques ». Les régions du cortex cérébral peuvent apprendre à représenter des choses totalement nouvelles en réaction à des changements divers. Si une partie du cortex est endommagé, d'autres parties vont s'adapter pour reprendre le rôle de la partie défaillante. Si un organe sensoriel est endommagé ou modifié, les régions associées du cortex s'adapteront pour représenter quelque chose d'autre. Le système est auto-ajustable.

Nous voulons que notre région HTM montre la même flexibilité. Si nous allouons 10000 colonnes à une région, elle devra apprendre comment représenter au mieux ses entrées avec ces 10 000 colonnes. Idem si nous lui en allouons 20 000. Si la répartition statistiques des entrées changent au cours du temps, les colonnes devront changer elles aussi pour représenter au mieux cette nouvelle réalité. Pour faire court, le concepteur d'une HTM devrait être en mesure d'allouer n'importe quelles ressources à une région et cette dernière fera de son mieux pour représenter ses entrées en fonction du nombre de colonnes et de la répartition statistiques des entrées. La règle générale veut qu'avec plus de colonnes dans une région, chaque colonne représentera des motifs d'entrée plus grands et plus détaillés et qu'elle sera aussi active moins fréquemment tout en maintenant un niveau relativement constant de parcimonie.

Aucune nouvelle règle d'apprentissage n'est nécessaire pour atteindre cet objectif hautement souhaitable. En dopant les colonnes inactives, en inhibant les colonnes voisines pour maintenir un niveau de parcimonie constant, en établissant des seuils minimaux sur les entrées, en maintenant une large population de synapses potentiels et en ajoutant ou bien en oubliant les synapses en fonction de leur contribution, les colonnes se configureront dynamiquement pour atteindre le comportement voulu.

Détails du concentrateur spatial

Nous pouvons maintenant parcourir les fonctions qui sont assurées par le concentrateur spatial.

1) Démarrer avec une entrée constituée d'un nombre fixe de bits. Ces derniers peuvent représenter soit des données sensorielles, soit des données provenant de la sortie d'une autre région dans la hiérarchie.

2) Assigner un nombre fixe de colonnes à la région qui reçoit cette entrée. Chaque colonne a un segment de dendrites associé, lui-même pourvu d'un jeu de synapses potentielles représentant un sous ensemble des bits d'entrée. Chaque synapse potentielle possède une valeur de permanence qui selon sa valeur valide ou pas la synapse potentielle.

3) Pour toute entrée, déterminer combien de synapses valides dans chaque colonne sont connectées aux bits d'entrée actifs.

4) Le nombre de synapses actives est multiplié par un facteur d'amplification ("boosting" factor) qui est déterminé dynamiquement en fonction de la fréquence d'activation d'une colonne par rapport à ses voisines.

5) Les colonnes ayant l'activation la plus élevée après la stimulation désactivent toutes les colonnes dans le rayon d'inhibition sauf un pourcentage fixe d'entre elles. Le rayon d'inhibition est lui-même déterminé dynamiquement par l'étalement (« fan-out ») des bits d'entrée. Nous obtenons ainsi un ensemble parcimonieux de colonnes actives.

6) Pour chacune des colonnes actives nous ajustons les valeurs de permanence de toutes les synapses potentielles. Les valeurs de permanence des synapses alignées avec des bits d'entrée actifs sont incrémentées. Les valeurs de permanence des synapses alignées avec des bits d'entrée inactifs sont décrémentées. Les modifications effectuées sur les valeurs de permanence peuvent entrainer le changement d'état des synapses de actif à inactif et vice-versa.

Concepts du concentrateur temporel

Rappelez-vous que le concentrateur temporel apprend des séquences et fait des prédictions. La méthode de base s'appuie sur le fait que lorsqu'une cellule devient active, elle forme des connexions avec d'autres cellules activées juste avant.
Les cellules peuvent ensuite prédire leur activation en regardant leurs connexions. Si toutes les cellules procèdent de la sorte, elles peuvent ensemble stocker et se remémorer des séquences et prédire ainsi ce qui va probablement se produire dans le futur. Il n'existe pas de stockage central pour les séquences de motifs ; la mémoire est en fait distribuée sur chacune des cellules. C'est cette distribution de l'information qui rend le système robuste au bruit et aux erreurs. Une cellule peut ainsi échouer dans sa tâche et n'entrainer que peu ou pas d'effet discernable.

Il faut avoir présent à l'esprit quelques-unes des propriétés importantes des représentations distribuées parcimonieuses qui sont exploitées par le concentrateur temporel.

Supposons une région hypothétique formant des représentations en s'appuyant sur 200 cellules actives sur un total de 10 000 (soit 2% de cellules actives à tout moment). Comment pouvons-nous mémoriser et reconnaitre un motif particulier de 200 cellules actives ? Une façon simple consiste à faire une liste des 200 cellules en question. Si nous voyons les 200 mêmes cellules actives à nouveau nous pourrons reconnaitre le motif. Mais que se passerait-il si nous faisions seulement une liste de 20 cellules parmi les 200 actives et ignorions les 180 autres ? Vous pourriez penser que se souvenir uniquement de 20 cellules va causer beaucoup d'erreurs, et que ces 20 cellules pourraient se retrouver actives en même temps dans de très nombreux motifs de 200 cellules. En fait il n'en est rien. Comme les motifs sont de grande taille et parcimonieux (dans cet exemple uniquement 200 cellules actives sur 10 000), se souvenir de 20 cellules est quasiment aussi bien que d'en mémoriser 200. En pratique les chances d'erreur sont extrêmement faibles et nous avons au passage considérablement réduit l'espace mémoire nécessaire au stockage de l'information.

Les cellules d'une région HTM tirent parti de cette propriété. Chacun des segments de dendrites d'une cellule possède un ensemble de connexions vers d'autres cellules de la région. Un segment de dendrites forme ces connexions dans le but de reconnaitre l'état du réseau à un instant donné. Il se peut qu'il y ait des centaines ou des milliers de cellules actives à proximité mais le segment de dendrite n'a besoin de se connecter qu'à 15 ou 20 d'entre elles. Quand le segment de dendrite voit 15 de ces cellules actives, il peut être quasiment certain que le motif complet (de 200 cellules) est présent. Cette technique dite de sous –échantillonnage (sub-sampling) est omniprésente dans les algorithmes de la HTM.

Une cellule peut participer à de nombreux motifs distribués et dans de nombreuses séquences différentes. Une cellule particulière peut faire partie de dizaines ou centaines de transitions temporelles. C'est pour cette raison qu'une cellule possède

plusieurs segments de dendrites et non un seul. Idéalement une cellule devrait avoir un segment de dendrites pour chaque motif d'activité à reconnaitre. Cependant, en pratique, un segment de dendrites peut apprendre les connexions de plusieurs motifs totalement différents et continuer à fonctionner correctement. À titre d'exemple, un segment peut apprendre 20 connexions pour 4 motifs différents pour un total de 80 connexions. Nous fixons ensuite un seuil de sorte que le segment de dendrites devienne actif lorsque 15 de ses connexions deviennent actives. Cela laisse le champ libre à l'apparition d'erreurs. Il est possible que, par hasard, la dendrite atteigne son seuil de 15 connexions actives en mixant des morceaux de différents motifs. Cependant, ce type d'erreur est très improbable en raison, encore une fois, de la parcimonie des représentations.

Nous comprenons maintenant comment une unique cellule accompagnée d'une ou deux douzaines de segments de dendrites et quelques milliers de synapses peut reconnaitre des centaines de motifs d'activités des cellules.

Détails du concentrateur temporel

Nous énumérons ici les étapes parcourues par le concentrateur temporel. Nous reprenons là où le concentrateur spatial s'était arrêté, c'est-à-dire avec un ensemble de colonnes actives représentant l'entrée amont.

1) Pour chaque colonne active, identifier les cellules de la colonne en état prédictif et les activer. Si aucune cellule ne se trouve en état prédictif, activer toutes les cellules de la colonne. L'ensemble des cellules actives résultant constituent la représentation de l'entrée dans le contexte des entrées précédentes.

2) Pour chaque segment de dendrites de chaque cellule de la région, compter combien de synapses établies sont connectées à des cellules actives. Si ce nombre excède un certain seuil, ce segment est marqué comme actif. Les cellules avec un ou plusieurs segments de dendrites actifs sont placées en état prédictif sauf si elles sont déjà actives en raison d'une entrée aval (feed-forward input). Les cellules sans dendrite active et non activée en raison d'une entrée ascendante (bottom-up input) deviennent ou restent inactives. La collection des cellules maintenant en état prédictif constitue la prédiction faite par cette région.

3) Quand un segment de dendrites devient actif, modifier les valeurs de permanence de toutes les synapses associées à ce segment. Pour chaque synapse potentielle du segment de dendrites actif, augmenter la permanence des synapses connectées à des cellules actives et réduire la permanence des synapses connectées à des cellules inactives. Ces modifications apportées à la valeur de permanence des synapses sont marquées comme temporaires.

Ceci entraine la modification des synapses sur les segments qui sont déjà suffisamment entrainés pour rendre le segment actif et ainsi mener à une prédiction. Toutefois, dans la mesure du possible nous souhaitons étendre la fenêtre de temps de la prédiction le plus possible. C'est pour cette raison que nous choisissons un second segment de dendrites sur la cellule à entrainer. Pour ce second segment nous choisissons celui qui correspond le mieux à l'état du système dans l'étape de temps précédente. Pour ce segment, l'utilisation du l'état du système dans l'étape de temps précédente, accroit la permanence des synapses qui sont connectées à des cellules actives et décroit la permanence de celles qui sont connectées à des cellules inactives. Ces modifications apportées à la valeur de permanence des synapses sont marquées comme temporaires.

4) Lorsqu'une cellule bascule de l'état inactif à l'état actif en raison d'une entrée aval, nous parcourons chaque synapse potentielle associée à une cellule et supprimons toutes les marques temporaires. Ainsi nous mettons à jour la permanence des synapses uniquement si elles ont prédit correctement l'activation aval de la cellule.

5) Lorsqu'une cellule bascule de l'état actif à l'état inactif, défaire tout changement sur la permanence marqués comme temporaire pour chaque synapse potentiel de cette cellule. Nous ne voulons pas renforcer la permanence des synapses qui ont prédit l'activation aval d'une cellule de façon incorrecte.

Notez que seules les cellules qui sont actives en raison d'une entrée aval propage leur activité *à l'intérieur* de la région, sinon les prédictions amènerait encore plus de prédictions. Mais toutes les cellules actives (soit par entrée aval, soit prédictive) forme la sortie de la région et se propagent à la *prochaine* région de la hiérarchie.

Séquences et prédictions de premier ordre ou d'ordre variable

Il est un sujet que nous devons aborder avant de terminer notre discussion sur les concentrateurs temporel et spatial. Ce sujet n'est pas forcément intéressant pour tous les lecteurs et il n'est pas nécessaire à la bonne compréhension des chapitres 3 et 4.

Quel est l'impact lorsqu'on fait varier le nombre de cellules dans une colonne ? En particulier que se passe-t-il s'il n'y a qu'une seule cellule par colonne ?

Dans l'exemple utilisé plus haut, nous avons montré que la représentation d'une entrée comprenant 100 colonnes actives avec 4 cellules par colonne peut être encodée de 4^100 façons différentes. Ainsi, la même entrée peut apparaitre dans un grand nombre de contextes sans engendrer de confusion. Par exemple, si le motif d'entrée représente des mots, une région peut alors mémoriser de nombreuses phrases distinctes comportant sans cesse les mêmes mots sans qu'elles se

confondent. Un mot fréquemment répété tel que « chien » aurait ainsi une représentation unique dans ses différents contextes. Cette propriété confère à une région HTM la faculté de faire ce qu'on appelle des prédictions d' « ordre variable ». Une prédiction d'ordre variable n'est pas uniquement fondée sur ce qui est en train de se passer mais sur un nombre variable de contextes passés. Une région HTM est une mémoire d'ordre variable.

Si nous augmentons le nombre de cellules à cinq par colonne, le nombre d'encodages possibles pour une entrée donnée passera dans notre exemple à 5^100, un nombre considérablement plus élevé que 4^100. Chacun de ces nombres est si grand que pour la plupart des problèmes pratiques cette augmentation de capacité n'aura probablement aucune utilité.

Toutefois, diminuer considérablement le nombre de cellules par colonne a un impact considérable.

Si nous descendons jusqu'à une cellule par colonne, nous perdons la faculté à prendre en compte le contexte de nos représentations. C'est ainsi qu'une entrée donnée engendrera toujours la même prédiction quelle que soit l'activité passée. Avec une cellule par colonne, la mémoire d'une région HTM est dite de « premier ordre » ; les prédictions ne sont basées que sur l'état courant d'une entrée.

Les prédictions de premier ordre sont parfaitement adaptées pour un type de problème traité par le cerveau : l'inférence spatiale statique. Comme indiqué précédemment, un être humain exposé très brièvement à une image peut reconnaitre l'objet qui s'y trouve même si le laps de temps est trop court pour que l'œil ait le temps de bouger. Avec l'ouïe par contre, vous avez toujours besoin d'entendre une séquence de motifs sonores pour reconnaitre un morceau. La vision marche aussi de cette façon en traitant habituellement un flux d'images mais dans certaines conditions vous êtes capable de reconnaitre une image en une seule exposition.

Ces deux processus de reconnaissance statique d'une part et temporelle d'autre part pourraient donc nécessiter des mécanismes d'inférence distincts. L'un nécessite de reconnaitre des séquences de motifs dans un contexte de longueur variable. L'autre impose de reconnaitre un motif spatial statique sans l'aide d'un contexte temporel. Une région HTM à plusieurs cellules par colonne est parfaitement adaptée à la reconnaissance de séquences temporelles et une région à une cellule par colonne des séquences spatiales. Chez Numenta , nous avons réalisé de nombreuses expérimentations dans le domaine de la vision impliquant des régions à une seule cellule par colonne. Le détail de ses expérimentations déborde du cadre de ce chapitre mais nous allons néanmoins couvrir les points clés.

Si nous exposons une région HTM à des images, les colonnes de cette région vont apprendre à représenter des arrangements spatiaux de pixels communs aux

différentes images. Le type de motifs appris est très similaire à ce qu'on observe dans la région V1 du cortex cérébral (une région du cerveau particulièrement bien étudiée en biologie), à savoir des lignes et des angles des différentes orientations. En travaillant sur des images en mouvement la région HTM apprend aussi les transitions temporelles entre ces différentes formes. Par exemple, une ligne verticale dans une position donnée est souvent suivie d'une autre ligne verticale décalée à droite ou à gauche. Toutes ces transitions de motifs fréquemment observées sont mémorisées par la région HTM.

Maintenant que se passe-t-il si nous exposons une région à une scène montrant une ligne verticale se décalant vers la droite ? Si notre région ne possède qu'une seule cellule par colonne il en sortira une prédiction indiquant que la prochaine position sera à droite ou à gauche. En effet ne pouvant utiliser de contexte temporel la région sera incapable de dire dans quelle direction précise se déplace la ligne. Nous voyons en fait que les cellules à une seule colonne se comportent comme les « cellules complexes » du cortex cérébral. La sortie prédite par une cellule de ce type sera active pour une ligne en mouvement à droite ou à gauche ou bien statique. Nous avons aussi observé qu'une région de ce type montre une stabilité dans ses prédictions même si la scène subit des changements comme une translation, un changement d'échelle, etc… et ceci tout en maintenant sa capacité à reconnaitre des images comportant des objets de natures différentes. Ce comportement est exactement ce dont nous avons besoin pour reconnaitre l'invariance spatiale c'est-à-dire la capacité à reconnaitre le même motif à différents endroits d'une image.

Si maintenant nous conduisons la même expérimentation avec plusieurs cellules par colonne, nous verrons que les cellules se comportent comme des « cellules complexes directionnelles » du cortex cérébral. La sortie prédit par ces cellules ne seront actives que si une ligne se déplace spécifiquement à droite ou bien spécifiquement à gauche mais jamais pour les deux.

En rassemblant toutes ces observations, nous pouvons faire les hypothèses suivantes. Le cortex cérébral doit être en mesure de faire à la fois des prédictions de premier et d'ordre variable. Il existe 4 ou 5 couches de cellules dans chaque région du cortex cérébral. Ces couches diffèrent par plusieurs aspects mais elles partagent toutes la faculté d'organiser leurs réponses en colonnes et elles présentent une importante connectivité horizontale entre elles. Nous pensons que chaque couche de cellule du cortex cérébral effectue une variante des règles d'apprentissage et d'inférence de la HTM. Les différentes couches remplissent des rôles distincts. D'après les études anatomiques, on sait par exemple que la couche 6 crée un retour d'information (feedback) dans la hiérarchie et que la couche 5 est impliquée dans les comportements moteurs. Les deux principales couches de neurones ayant une action aval (feed-forward layers) sont les couches 3 et 4. Nous pensons que l'une des différences entre les couches 3 et 4 tient au fait que les cellules de la couche 4 agit indépendamment c'est-à-dire à une cellule par colonne alors que dans la couche 3 elles agissent à plusieurs. Ainsi les régions du cortex cérébral proches des entrées sensorielles possèdent une mémoire de premier ordre et aussi d'ordre variable. La

mémoire de séquences de premier ordre (correspondant grosso-modo à la couche de neurones IV) est utile à la formation de représentations qui restent invariantes malgré les changements spatiaux. La mémoire de séquences d'ordre variable (correspondant à peu près à la couche de neurones 3) est utile aux inférences et à la prédiction d'images en mouvement.

En résumé, nous faisons l'hypothèse que des algorithmes très proches de ceux décrits dans ce chapitre sont à l'œuvre dans toutes les couches de neurones du cortex cérébral. Les couches du cortex sont suffisamment différentes pour remplir des fonctions distinctes comme une action aval (feed-forward) ou amont (feedback), la focalisation de l'attention ou le comportement moteur. Dans les régions proches des entrées sensorielles, il est utile de disposer d'une couche de neurones proposant une mémoire de premier ordre permettant ainsi d'arriver à une invariance spatiale.

Chez Numenta, nous avons mené des expérimentations sur la reconnaissance d'images avec des régions HTM de premier ordre (une cellule par colonne). Nous avons aussi expérimenté des régions HTM d'ordre variable (plusieurs cellules par colonne) capable de reconnaitre et de prédire des séquences d'ordre variable. Il serait maintenant logique de combiner ces deux types de régions afin d'étendre les algorithmes à d'autres fins. Toutefois, nous pensons que de nombreux problèmes intéressants peuvent être résolus avec des régions à plusieurs cellules par colonne soit prises isolément soit organisées de façon hiérarchique.

Chapitre 3: Implémentation et pseudocode du concentrateur spatial

Ce chapitre présente le pseudocode détaillé d'une première implémentation du concentrateur spatial. L'entrée fournie au code est un tableau de valeurs binaires provenant de données sensorielles ou bien d'une région HTM de niveau inférieur. Le code calcule activeColumns(t) – soit la liste des colonnes gagnantes étant données les valeurs en entrée à l'instant t. Cette liste est ensuite fournie en entrée à la routine du concentrateur temporel décrite au chapitre suivant. En d'autres termes, activeColumns(t) est la sortie produite par la routine du concentrateur spatial.

Le pseudocode est découpé en trois phases distinctes se déroulant dans cet ordre :

Phase 1 : calculer le recouvrement avec l'entrée courante pour chaque colonne
Phase 2 : calculer les colonnes gagnantes après inhibition
Phase 3 : mettre à jour la valeur de permanence des synapses et les variables internes

Bien que la capacité d'apprentissage du concentrateur spatial soit mise en œuvre naturellement durant le processus, vous pouvez la désactiver en omettant la phase 3.

Le reste du chapitre présente le pseudocode pour chacune des 3 étapes. Les diverses structures de données et les routines auxiliaires sont présentées à la fin.

Initialisation

Avant de recevoir une entrée quelconque, la région est initialisée en calculant une liste initiale de synapses potentielles dans chaque colonne. Pour ce faire on choisit un ensemble aléatoire parmi l'espace des entrées. Chaque entrée est représentée par une synapse à laquelle on assigne une valeur de permanence aléatoire choisie selon deux critères. Tout d'abord, les valeurs sont choisies dans un petit intervalle autour de la valeur connectedPerm (c'est-à-dire la valeur de permanence au-delà de laquelle la synapse est considérée comme « connectée »). Cela permet à des synapses potentielles de devenir connectées (ou déconnectées) après un petit nombre d'itérations d'apprentissage. Ensuite, chaque colonne possède un centre naturel sur la région en entrée et les valeurs de permanence présentes une inclination naturelle vers ce centre (en pratique cela veut dire que les valeurs sont plus hautes près du centre).

Phase 1 : Recouvrement

Etant donné un vecteur d'entrée, la première phase consiste à calculer le recouvrement (overlap) de chaque colonne avec ce vecteur. Pour chaque colonne, le recouvrement en question est simplement le nombre de synapses ayant des entrées actives multiplié par son facteur d'amplification (boost). Si la valeur résultante est inférieure à minOverlap, le score de recouvrement est positionné à zéro.

```
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)
```

Phase 2 : Inhibition

La seconde phase permet de déterminer quelles sont les colonnes vainqueurs après l'étape d'inhibition. Le paramètre desiredLocalActivity contrôle le nombre de colonnes vainqueurs. Ainsi, si desiredLocalActivity vaut 10, une colonne figurera parmi les vainqueurs si son score de recouvrement est plus grand que le score de la 10e colonne la plus haute située dans son rayon d'inhibition.

```
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.
```

Phase 3 : Apprentissage

La troisième phase assure l'apprentissage ; elle met à jour des valeurs de permanence de toutes les synapses ainsi que le facteur d'amplification (boost) et le rayon d'inhibition.

La principale règle d'apprentissage se situe entre les lignes 20 et 26. Pour les colonnes vainqueurs , si une synapse est active sa valeur de permanence est incrémentée et décrémentée dans le cas contraire. Les valeurs de permanence sont bornées entre 0 et 1.

Les lignes 28 à 36 mettent en œuvre l'amplifaction (boosting). Il existe 2 mécanismes d'amplification distincts permettant à une colonne de développer de nouvelles connexions. Si une colonne ne gagne pas assez souvent (tel que défini par activeDutyCycle), son facteur d'amplification global est augmenté (ligne 30 à 32). Par ailleurs, si les synapses connectées d'une colonne ne recouvrent correctement aucune des entrées de façon suffisamment fréquente (tel que défini par overlapDutyCycle), toutes ses valeurs de permanence sont alors amplifiées (ligne 34 à 36). Note : une fois l'apprentissage arrêté, la variable boost(c) est figée.

Enfin, pour terminer la phase 3, le rayon d'inhibition est recalculé (ligne 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.
```

Structures de données et routines

Les variables et structures de données suivantes sont utilisées dans notre pseudocode :

columns	Liste des colonnes.
input(t,j)	L'entrée au niveau j à l'instant t. input(t, j) est à 1 si l'entrée de rang j est active (on).
overlap(*c*)	Le recouvrement de la colonne c du concentrateur spatial avec une entrée particulière.
activeColumns(*t*)	Liste des indices des colonnes vainqueurs suite aux entrées ascendantes (bottom-up input).
desiredLocalActivity	Paramètre permettant de contrôler le nombre de colonnes vainqueurs après l'étape d'inhibition.
inhibitionRadius	Taille moyenne du champ réceptif connecté des colonnes.
neighbors(c)	Une liste de toutes les colonnes dans un rayon inhibitionRadius de la colonne c.
minOverlap	Le nombre minimum d'entrées qui doivent être actives pour qu'une colonne soit prise en compte dans la phase d'inhibition.
boost(c)	La valeur du facteur d'amplification de la colonne c telle que calculée durant la phase d'apprentissage – utilisée pour accroitre la valeur de recouvrement des colonnes inactives.
synapse	Structure de données représentant une synapse – contient une valeur de permanence et l'index de la source en entrée.
connectedPerm	Si la valeur de permanence d'une synapse est supérieure à cette valeur, elle est dite connectée.
potentialSynapses(c)	La liste des synapses potentielles et leurs valeurs de permanence.
connectedSynapses(c)	Un sous-ensemble de potentialSynapses(c) pour lequel la valeur de permanence est supérieure à connectedPerm. Elles représentent les entrées ascendantes (bottom-up inputs) connectées à la colonne c à un moment donné.
permanenceInc	L'incrément appliqué aux valeurs de permanence des synapses durant la phase d'apprentissage.

permanenceDec	Le décrément appliqué aux valeurs de permanence des synapses durant la phase d'apprentissage.
activeDutyCycle(c)	La moyenne mobile représentant le nombre de fois qu'une colonne c a été active après la phase d'inhibition (par exemple sur les 1000 dernières itérations).
overlapDutyCycle(c)	La moyenne mobile représentant le nombre de fois qu'une colonne c a eu un recouvrement significatif (par exemple plus grand que minOverlap) avec ses entrées (par exemple sur les 1000 dernières itérations).
minDutyCycle(c)	Une variable représentant le taux d'activation minimum souhaité pour une cellule. Si le taux tombe sous ce seuil, la cellule sera amplifiée. Cette valeur est égale à 1% du taux d'activation maximum de ses voisines.

Les routines suivantes sont utilisées dans le pseudocode ci-dessus.

kthScore(cols, k)

Etant donnée la liste des colonnes, retourne la k'ième plus haute valeur de recouvrement.

updateActiveDutyCycle(c)

Calcule la moyenne mobile du nombre de fois que la colonne c a été active après la phase d'inhibition.

updateOverlapDutyCycle(c)

Calcule la moyenne mobile du nombre de fois que la colonne c a eu une valeur de recouvrement plus grand que minOverlap.

averageReceptiveFieldSize()

Calcule le rayon de la taille moyenne du champ réceptif connecté de toutes les colonnes. La taille du champ réceptif connecté d'une colonne inclut uniquement les synapses connectées (celles dont la valeur de permanence est supérieure ou égale à connectedPerm). Cette valeur est utilisée pour déterminer le champ d'inhibition latéral entre colonnes.

maxDutyCycle(cols)

Retourne le maximum de cycles d'utilisation actifs parmi la liste des colonnes fournies en paramètre.

increasePermanences(c, s)

Incrémente la valeur de permanence de toutes les synapses de la colonne c d'un facteur s.

boostFunction(c)

Retourne le facteur d'amplification de la colonne c. Il s'agit d'une valeur scalaire >= 1. Si activeDutyCyle(c) est supérieur à minDutyCycle(c), la valeur est 1. La valeur d'amplification croit linéairement dès que le paramètre activeDutyCyle de la colonne tombe sous la valeur en dessous de minDutyCycle.

Chapitre 4: Implémentation et pseudocode du concentrateur temporel

Ce chapitre présente le pseudocode détaillé d'une première implémentation de la fonction concentrateur temporel. activeColumns(t) tel que calculé par le concentrateur spatial est fourni en entrée à ce code. Il calcule les états actifs et prédictifs pour chaque cellule au pas de temps courant t. Un OU booléen des états actifs et prédictifs de chaque cellule constitue la sortie du concentrateur temporel pour le niveau suivant.

Le pseudocode se compose de trois phases distinctes exécutées en séquence :

Phase 1 : calculer l'état actif, activeState(t), pour chaque cellule
Phase 2 : calculer l'état prédictif, predictiveState(t), pour chaque cellule
Phase 3 : mettre à jour les synapses

La phase 3 n'est nécessaire que pour l'apprentissage. Cependant, contrairement au concentrateur spatial, les phases 1 et 2 comportent certaines étapes spécifiques à l'apprentissage lorsque celui-ci est actif. Etant donné que le concentrateur temporel est nettement plus complexe que le concentrateur spatial, nous montrons d'abord la version du concentrateur temporel basé uniquement sur l'inférence puis la version combinant inférence et apprentissage. Une description de quelques détails d'implémentation, de terminologie et des routines auxiliaires sont présentés après le pseudocode en fin de chapitre.

Pseudocode du concentrateur temporel : inférence seule

Phase 1

La première phase calcule l'état actif pour chaque cellule. Pour chaque colonne vainqueur nous déterminons quelles cellules vont devenir actives. Si l'entrée ascendante (bottom-up input) a été prédite par une cellule (c'est-à-dire que son predictiveState était à 1 en raison d'un segment de séquences de l'étape précédente), alors ces cellules deviennent actives (lignes 4 à 9). Si l'entrée ascendante était inattendue (c'est-à-dire qu'aucune cellule n'avait sa sortie predictiveState active), alors chaque cellule de la colonne devient active (lignes 11 à 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
```

Phase 2

La seconde phase calcule l'état prédictif pour chaque cellule. Une cellule activera son predictiveState si l'un quelconque de ses segments devient actif c'est-à-dire qu'un nombre suffisant de connexions horizontales sont allumées en raison d'entrées avals (feed-forward input).

```
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
```

Pseudocode du concentrateur temporel : inférence et apprentissage combinés

Phase 1

La première phase calcule activeState pour chaque cellule d'une colonne vainqueur. Pour ces colonnes, le code sélectionne en plus une cellule par colonne comme la cellule d'apprentissage (learnState). La logique est la suivante : si l'entrée ascendante (botton-up input) était déjà prédite par l'une des cellules (c'est-à-dire sa sortie predictiveState était à 1 en raison d'un segment de séquences), alors ces cellules deviennent actives (ligne 23 à 27). Si ce segment est devenu actif à partir de cellules choisies pour leur learnState actif, cette cellule est sélectionnée comme cellule d'apprentissage (lignes 28 à 30). Si l'entrée ascendante (bottom-up input) n'était pas prédite alors toutes les cellules du segment deviennent actives (lignes 32 à 34). De plus, la cellule la plus appropriée est choisie comme cellule d'apprentissage (lignes 36 à 41) et un nouveau segment est ajouté à cette cellule.

```
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)
```

Phase 2

La seconde phase calcule l'état prédictif de chaque cellule. Une cellule activera son état prédictif si l'un de ses segments devient actif, c'est-à-dire si un nombre suffisant de ses entrées latérales sont actives suite à des entrées avals (feed-forward input). Dans ce cas, la cellule met en file d'attente les modifications suivantes : a) renforcement du segment actuellement actif (lignes 47 à 48) et b) renforcement d'un segment qui aurait pu prédire cette activation, c'est-à-dire un segment qui a une correspondance (potentiellement faible) avec l'activité observée durant l'étape de temps précédente (lignes 50 à 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)
```

Phase 3

La troisième et dernière phase se charge de l'apprentissage. Dans cette phase les modifications des segments mis en fil d'attente sont effectuées une fois l'entrée aval (feed-forward input) disponible et la cellule est choisie comme cellule d'apprentissage (lignes 56 à 57). Sinon, si la cellule cesse de prédire correctement pour une raison quelconque, les segments sont renforcés négativement (lignes 58 à 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.
```

Détails d'implémentation et terminologie

Dans cette section nous décrivons quelques-uns des détails de notre implémentation et de la terminologie de notre concentrateur temporel. Chaque cellule est indexée en utilisant deux nombres : un index de colonne c et un index de cellule i. Les cellules entretiennent une liste de segments dendritiques où chaque segment contient une liste de synapses ainsi qu'une valeur de permanence pour chaque synapse. Les changements opérés sur une synapse de cellule sont marqués comment temporaires jusqu'à ce que la cellule devienne active en raison d'entrées aval (feed-forward input). Ces changements temporaires sont conservés dans segmentUpdateList. Chaque segment maintient aussi une valeur booléenne, sequenceSegment, indiquant si le segment prédit une entrée aval (feed-forward input) au prochain pas de temps.

L'implémentation de synapses potentielles est différente de l'implémentation vue précédemment dans le concentrateur spatial. Dans ce dernier, la liste complète des synapses potentielles est représentée sous forme d'une liste explicite. Dans le concentrateur temporel, chaque segment peut posséder sa propre liste de synapses (possiblement de grande taille). En pratique maintenir une longue liste pour chaque segment est couteux à la fois en espace mémoire et en temps de calcul. C'est pourquoi dans le concentrateur temporel, nous ajoutons aléatoirement des synapses actives à chaque segment durant la phase d'apprentissage (contrôlé par le paramètre newSynapseCount). Cette optimisation produit un effet identique à celui d'une liste complète de synapses potentielles mais la liste par segment est beaucoup plus petite tout en maintenant la possibilité d'apprendre de nouveaux motifs temporels.

Le pseudocode utilise une petite machine d'états pour garder la trace de l'état des cellules à différents pas de temps. Trois états sont maintenus pour chaque cellule. Les tableaux activeState et predictiveState conservent les états actifs et prédictifs de chaque cellule à chaque pas de temps. Le tableau learnState détermine quelles sorties de cellules sont utilisées pour l'apprentissage. Lorsqu'une entrée est inattendue toutes les cellules d'une colonne donnée deviennent actives dans le même pas de temps. Une seule de ces cellules (celle qui correspond le mieux à l'entrée) voit son learnState activé. Nous ajoutons uniquement des synapses aux cellules dont le learnState est à 1 (ceci évite de surreprésenter une colonne totalement active dans les segments dendritiques).

Les structures de données suivantes sont utilisées dans le pseudocode du concentrateur temporel :

cell(c,i)	La liste de toutes les cellules indexée par *i* et *c*.
cellsPerColumn	Nombre de cellules dans chaque colonne.
activeColumns(*t*)	Liste des index des colonnes vainqueurs suite à une entrée aval (botton-up input) (il s'agit de la sortie du concentrateur spatial).
activeState(c, i, t)	Un vecteur de booléens avec un seul nombre par cellule. Il représente l'état actif de la colonne c, cellule i à l'instant t étant donnée l'entrée aval (feed-forward input) et le contexte temporel passé. activeState(c, i, t) est la contribution de la colonne c, cellule i à l'instant t. Si sa valeur est à 1, la cellule a une entrée aval à l'état courant ainsi qu'un contexte temporel correct.
predictiveState(c, i, t)	Un vecteur de booléens avec un seul nombre par cellule. Il représente l'état prédictif de la colonne c, cellule i à l'instant t étant donnée l'activité ascendante (bottom-up activity) des autres colonnes et le contexte temporel passé. predictiveState(c, i, t) est la contribution de la colonne c, cellule i à l'instant t. Si sa valeur est à 1, la cellule prédit l'entrée aval (feed-forward input) dans le contexte temporel courant.
learnState(c, i, t)	Un booléen indiquant si la cellule i de la colonne c est retenue comme cellule d'apprentissage.
activationThreshold	Seuil d'activation pour un segment. Si le nombre de synapses connectées actives dans un segment est supérieur à activationThreshold, the segment est réputé actif.
learningRadius	L'aire autour d'une cellule du concentrateur temporel dans laquelle une connexion latérale peut être établie.
initialPerm	Valeur de permanence initiale d'une synapse.
connectedPerm	Si la valeur de permanence d'une synapse est supérieure à cette valeur elle est réputée connectée.
minThreshold	Activité minimum d'un segment pour l'apprentissage.
newSynapseCount	Le nombre maximum de synapses ajoutées à un segment durant l'apprentissage.

permanenceInc	La valeur d'incrément de la valeur de permanence des synapses en phase d'apprentissage.
permanenceDec	La valeur de décrément de la valeur de permanence des synapses en phase d'apprentissage.
segmentUpdate	Structure de données contenant trois éléments d'information nécessaire à la mise à jour d'un segment : a) l'index du segment (-1 s'il est nouveau), b) la liste des synapses actives existantes et c) un flag indiquant si le segment doit être considéré comme un segment de séquence (valeur par défaut : **false**).
segmentUpdateList	Une liste de structures de type segmentUpdate structures. segmentUpdateList(c,i) stocke la liste des changements pour la cellule d'index i dans la colonne c.

Les routines suivantes sont utilisées dans le code ci-dessus :

segmentActive(s, t, state)

Cette routine renvoie **true** (vrai) si le nombre de synapses connectées sur le segment s qui sont actifs en raison de l'état au pas de temps t est plus grand que activationThreshold. Le paramètre state peut être activeState ou learnState.

getActiveSegment(c, i, t, state)

Etant donné la colonne c cellule i, retourne un index de segment tel que segmentActive(s,t, state) est à **true**. Si plusieurs segments sont actifs, les segments de séquence ont la préférence. Dans le cas contraire, les segments avec le plus d'activité ont la préférence.

getBestMatchingSegment(c, i, t)

Etant donné la colonne c cellule i, trouver le segment avec le plus grand nombre de synapses actives. Cette routine est assez agressive dans sa recherche de la meilleure correspondance. La valeur de permanence des synapses est autorisée à se situer au-dessous de connectedPerm. Le nombre de synapses actives peut se situer en dessous de activationThreshold mais doit rester au-dessus de minThreshold. La routine renvoie l'index du segment. Si aucun segment n'est trouvé, la valeur -1 est retournée.

getBestMatchingCell(c)

Etant donnée une colonne, renvoie la cellule avec le segment le plus approprié (selon les critères définis dans la routine précédente). Si aucune cellule n'a de segment correspondant alors la routine retourne la cellule avec le plus petit nombre de segments.

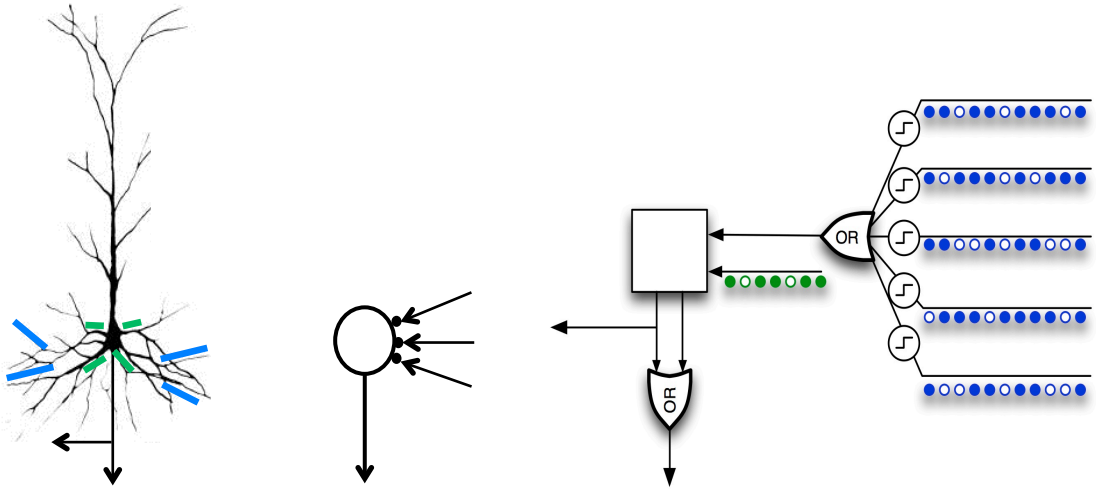
getSegmentActiveSynapses(c, i, t, s, newSynapses= **false**)

Retourne une structure de données segmentUpdate contenant la liste des modifications proposées sur le segment s. Soit activeSynapses la liste des synapses actives où les cellules d'origine ont leur activeState à 1 au pas de temps t (cette liste est vide si s = -1 puisque le segment n'existe pas). newSynapses est un argument optionnel dont la valeur par défaut est **false**. S'il est positionné à **true** alors newSynapseCount - count(activeSynapses) synapses sont ajoutées à activeSynapses. Ces synapses sont choisies aléatoirement parmi l'ensemble des cellules dont le paramètre learnState = 1 au pas de temps t.

adaptSegments(segmentList, positiveReinforcement)

Cette fonction itère sur la liste des segmentUpdate et renforce chaque segment. Pour chaque élément de segmentUpdate, les modifications suivantes sont effectuées. Si positiveReinforcement est à **true** alors les synapses dans la liste des synapses actives voient leur valeur de permanence incrémentée de permanenceInc. Toutes les autres synapses voient leur valeur de permanence décroitre de permanenceDec. Si positiveReinforcement est à **false** alors les synapses dans la liste des synapses actives voient leur valeur de permanence décroitre de permanenceDec. Après cette étape, toute synapse dans segmentUpdate qui existe encore voit sa valeur de permanence augmenter de initialPerm.

Annexe A : comparaison entre les neurones biologiques et les cellules HTM



L'image ci-dessus montre un neurone biologique à gauche, un neurone artificiel simple au centre et un neurone HTM ou « cellule » à droite. L'objectif de cette annexe est de vous donner une meilleure compréhension des cellules HTM et de leur fonctionnement en les comparant à de véritables neurones ainsi qu'à des neurones artificiels simples.

Les vrais neurones sont incroyablement compliqués et variés. Nous allons nous concentrer sur les grands principes généraux et uniquement sur ceux qui sont pertinents pour notre modèle. Bien que nous ignorions de nombreux détails des vrais neurones, les cellules utilisées dans les algorithmes d'apprentissage corticaux HTM sont bien plus réalistes que les neurones artificiels mis en œuvre dans la plupart des réseaux de neurones. Tous les éléments utilisés dans les cellules HTM sont nécessaires pour le bon fonctionnement d'une région HTM.

Les neurones biologiques

Les neurones sont les cellules porteuses d'information du cerveau. L'image de gauche est typique d'un neurone excitateur. L'apparence visuelle d'un neurone est dominée par les branches de dendrites. Toutes les entrées excitatrices d'un neurone se font via les synapses qui sont réparties le long des dendrites. Dans les dernières années notre connaissance des neurones a considérablement progressé. Le plus grand bond en avant s'est produit lorsque nous avons compris que les dendrites d'un neurone n'étaient pas uniquement des conduits amenant les signaux d'entrée

au corps de la cellule. Nous savons maintenant que les dendrites sont en eux-mêmes des éléments complexes et non-linéaires de traitement de l'information. Les algorithmes d'apprentissage corticaux HTM tirent parti de ces propriétés non linéaires.

Les neurones sont composés de plusieurs parties.

Le corps de la cellule
Le corps de la cellule est un petit volume au centre du neurone. La sortie de la cellule, l'axone, prend naissance au corps de la cellule. Les entrées de la cellule sont les synapses réparties le long des dendrites qui alimentent le corps de la cellule.

Dendrites proximales
Les branches dendritiques les plus proches du corps de la cellule sont appelées dendrites proximales. Sur la figure certaines d'entre elles sont signalées par une ligne verte.

De multiples synapses actives simultanément sur des dendrites proximales ont un effet à peu près additif sur le corps de la cellule. Ainsi 5 synapses actives amèneront un effet de dépolarisation environ 5 fois plus grand qu'une seule synapse active. A l'opposé, si une seule synapse est activée de façon répétée par une succession rapide de potentiels d'action, le deuxième, le troisième potentiel d'action et les suivants auront beaucoup moins d'effet sur le corps de la cellule que le premier.

Ainsi, nous pouvons dire que les entrées situées sur les dendrites proximales s'additionnent linéairement au niveau du corps de la cellule et que les impulsions neurales (neural spikes) arrivant sur une unique synapse auront un effet à peine supérieur à celui d'une seule impulsion.

Les connexions à action aval (feed-forward connections) vers une région du cortex cérébral se font préférentiellement via les dendrites proximales. Cette constatation a été faite au moins pour ce qui concerne la couche de neurones 4, la première couche d'entrée de chaque région du cerveau.

Dendrites distales
Les branches dendritiques plus éloignées du corps de la cellule sont appelées dendrites distales. Dans la figure ci-dessus elles sont marquées d'un trait bleu.

Les dendrites distales sont plus fines que les proximales. Elles se connectent à d'autres dendrites sur des branches de l'arbre dendritique et non pas directement sur le corps de la cellule. Ces différences donnent aux dendrites distales des propriétés électriques et chimiques spécifiques. Lorsqu'une seule synapse est activée sur une dendrite distale elle n'a qu'un effet minime sur le corps de la cellule. La dépolarisation qui apparait localement à la synapse s'affaiblit au fur et à mesure de sa progression vers le corps de la cellule. Pendant des années ce comportement

est resté un vrai mystère. Il semblait en effet que les synapses distales, qui représentent la majorité des synapses d'un neurone, n'avaient que peu d'effet.

On sait maintenant que les sections des dendrites distales agissent comme des régions de traitement semi-indépendantes. Si suffisamment de synapses deviennent actives au même moment sur un court segment de la dendrite, elles peuvent générer une impulsion dendritique qui peut progresser jusqu'au corps de la cellule avec un effet important. À titre d'exemple, vingt synapses actives dans un segment de 40 µm (microns) génère une impulsion dendritique.

Ainsi nous pouvons dire que les dendrites distales agissent comme un détecteur de coïncidence avec un seuil de déclenchement.

Les synapses des dendrites distales se forment prioritairement avec des cellules proches dans la même région.

La figure ci-dessus montre une grande branche dendritique s'allongeant vers le haut aussi appelée dendrite apicale. Une théorie avance que cette structure permet au neurone de placer plusieurs dendrites distales dans une zone où elles peuvent plus facilement établir des connexions avec des axones passant à proximité. Dans cette interprétation, la dendrite apicale agit comme une extension de la cellule.

Synapses

Un neurone typique peut avoir des milliers de synapses. Une grande majorité (peut-être 90%) d'entre elles se situe sur les dendrites distantes et le reste sur les dendrites proximales.

Pendant de nombreuses années on a supposé que l'apprentissage passait par le renforcement ou l'affaiblissement de l'effet ou "poids" des synapses. Bien que cet effet ait effectivement était observé chaque synapse se comporte de façon plus ou moins stochastique. Lorsqu'elle est activée elle ne libérera pas un neurotransmetteur de façon fiable. Ainsi les algorithmes utilisés par le cerveau ne peuvent pas se reposer sur la précision ou la fidélité des poids de synapses individuelles.

Qui plus est, nous savons aujourd'hui que des synapses se font et se défont rapidement. Cette flexibilité représente une forme puissante d'apprentissage et explique bien mieux l'acquisition rapide de connaissances. Une synapse ne se formera que si un axone et une dendrite se situent à proximité l'une de l'autre, amenant ainsi au concept de synapses « potentielles ». Avec ses hypothèses, l'apprentissage se ferait pour une large part en formant des synapses vraies à partir de synapses potentielles.

Sortie d'un neurone

La sortie d'un neurone se présente sous forme d'une impulsion électrique aussi appelée "potentiel d'action" qui se propage le long de l'axone. En quittant le corps de

la cellule l'axone se divise presqu'invariablement en deux. Une branche voyage à l'horizontal en établissant des connexions avec des cellules proches. L'autre branche se projette dans les autres couches de cellules ou ailleurs dans le cerveau. Dans l'image du neurone biologique ci-dessus l'axone n'était pas visible et nous avons donc ajouté une ligne et 2 flèches pour le représenter.

Bien que la sortie d'un neurone soit toujours une impulsion, il existe différentes façon de l'interpréter. L'opinion qui prévaut (en particulier pour ce qui concerne le cortex cérébral) est que la fréquence des impulsions est ce qui compte avant tout. Ainsi la sortie d'une cellule peut être vue comme une valeur scalaire.

Quelques neurones montrent aussi des "bouffées" d'activité, des séries courtes et rapides de quelques impulsions différentes du motif d'impulsion habituel.

Cette description du neurone est juste une brève introduction. Elle se concentre sur des propriétés qui correspondent aux caractéristiques des cellules HTM et laisse de côté de nombreux autres détails. Certaines propriétés décrites ici ne sont pas universellement admises. Nous les mentionnons car elles sont nécessaires à l'élaboration de notre modèle. Ce qu'on sait des neurones pourrait facilement remplir plusieurs ouvrages et la recherche sur les neurones reste encore aujourd'hui très active.

Neurones artificiels simples

L'image du centre au début de cette annexe montre un élément de type neurone utilisé dans nombre de réseaux de neurones artificiels classiques. Ces neurones artificiels possèdent un ensemble de synapses ayant chacune un poids. Chaque synapse reçoit une activation sous la forme d'une valeur scalaire qui est multipliée par le poids de la synapse. La sortie de toutes les synapses est additionnée de façon non linéaire pour produire la sortie du neurone artificiel. L'apprentissage se fait en ajustant les poids des synapses et parfois la fonction non linéaire.

Ce type de neurone artificiel, et ses variations, ont prouvé leur utilité dans nombre d'application en tant qu'outil computationnel. Cependant, il ne rend que très peu compte de la complexité et de la puissance de traitement des neurones biologiques. Si nous voulons comprendre et modéliser comment un ensemble de neurones travaillent dans le cerveau nous avons besoins d'un modèle de neurones nettement plus élaboré.

Les cellules HTM

Dans notre figure, l'image sur la droite décrit une cellule utilisée dans les algorithmes d'apprentissage cortical HTM. Une cellule HTM traduit un bon nombre

des caractéristiques importantes des vrais neurones mais procède aussi à quelques simplifications.

Dendrite Proximale
Chaque cellule HTM possède une seule dendrite proximale. Toutes les entrées aval (feed-forward input) vers la cellule se font via des synapses (ici représentées en vert). L'activité des synapses est sommée linéairement pour produire une activation aval (feed-forward activation) pour la cellule.

Nous imposons à toutes les cellules d'une colonne de fournir la même réponse aval (feed-forward response). Dans un vrai neurone ce serait très certainement fait par une sorte de cellule inhibitrice. Dans les MTHs nous forçons simplement toutes les cellules d'une colonne à partager une dendrite proximale unique.

Pour éviter les cellules qui ne gagneraient jamais dans leur compétition avec les voisines, une cellule HTM amplifiera leur activation aval (feed-forward activation) si elles ne gagnent pas assez souvent relativement à leur voisines. Cette compétition n'est pas illustrée dans le diagramme.

Enfin, la dendrite proximale possède un ensemble de synapses potentielles qui représentent un sous-ensemble de toutes les entrées d'une région. Au fur et à mesure de leur apprentissage les cellules augmentent ou réduisent la valeur de « permanence » de toutes les synapses potentielles de la dendrite proximale. Seules celles qui dépassent un certain seuil deviennent valides.

Comme mentionné plus haut, le concept de synapses potentielles nous vient directement de la biologie et fait référence aux axones et dendrites suffisamment proches l'un de l'autre pour former des synapses. Nous étendons ce concept à un ensemble plus large de connexions potentielles pour la cellule HTM. Dendrites et axones d'un neurone biologique peuvent croitre et décroitre lors de l'apprentissage et ainsi faire varier l'ensemble des synapses potentielles. En assignant d'emblée à chaque cellule un grand ensemble de synapses potentielles nous arrivons à peu près au même résultat. Les synapses potentielles ne sont pas montrées sur le diagramme.

En combinant la compétition entre les colonnes, l'apprentissage à partir d'un jeu de synapses potentielles et l'amplification des colonnes sous-utilisées nous donnons à une région de neurones HTM une grande plasticité aussi observée dans le cerveau. Une région HTM ajustera ainsi automatiquement ce que représente chaque colonne (via les modifications opérées sur les synapses potentielles) si l'entrée change ou si le nombre de colonnes croit ou diminue.

Dendrite Distales
Chaque cellule HTM maintient une liste de segments de dendrites distales. Chaque segment agit comme un détecteur de seuil. Si le nombre de synapses actives sur un segment (indiqué en bleu sur la figure) dépasse le seuil, le segment devient actif et la

cellule associée passe à l'état prédictif. L'état prédictif d'une cellule est calculé via un OU booléen de l'état d'activation de tous ses segments.

Un segment dendritique mémorise l'état de la région en formant des connexions vers des cellules qui étaient toutes actives au même instant. Ce segment se souvient de l'état avant que la cellule ne devienne active en raison d'une entrée aval (feed-forward input). Ainsi les segments recherchent un état qui prédit que sa cellule va devenir active. Une valeur de seuil typique pour un segment dendritique est de 15. Si 15 synapses valides sont actives simultanément, la dendrite devient active. Il peut y avoir des centaines ou des milliers de cellules actives autour mais la connexion à 15 d'entre elles est suffisante pour reconnaitre le motif d'ensemble.

Chaque dendrite distale possède aussi un ensemble de synapses potentielles. Elles représentent un sous-ensemble de toutes les cellules d'une région. Lors de son apprentissage, le segment augmente ou diminue la valeur de permanence de toutes ses synapses potentielles. Seules les synapses potentielles qui dépassent un certain seuil deviennent valides.

Dans une première implémentation nous avons utilisé un nombre fixe de segments dendritiques par cellule. Dans une autre nous avons ajouté ou supprimé des segments au fur et à mesure de l'apprentissage. Les deux méthodes fonctionnent. Si le nombre de segments par cellule est fixe, il est possible de stocker différents ensemble de synapses sur le même segment. Par exemple, avec 20 synapses valides sur un segment et un seuil de 15 (en général le seuil doit être inférieur au nombre de synapses pour être insensible au bruit), le segment peut reconnaitre un état particulier des cellules alentour. Mais que se passerait-il si nous ajoutions un autre jeu de 20 synapses au segment représentant un état des cellules proches totalement différent ? Cela introduit une possibilité d'erreur car le segment pourrait avoir 8 synapses actives dans un motif et 7 dans l'autre et ainsi devenir actif à tort. Nous avons déterminé expérimentalement qu'un maximum de 20 motifs peuvent être stockés sur un segment avant que des erreurs n'apparaissent. Une cellule HTM pourvue d'une douzaine de segments dendritiques peut donc s'impliquer dans un nombre de prédictions distinctes.

Synapses

Les synapses d'une cellule HTM ont un poids binaire. Rien n'empêche le modèle HTM d'utiliser une valeur scalaire pour le poids mais en raison de l'utilisation de motifs distribués de façon parcimonieuse nous n'avons pas eu besoin de recourir aux poids scalaires.

Cependant, les synapses d'une cellule HTM possède une valeur scalaire appelée "permanence" qui est ajustée pendant l'apprentissage. Une valeur de permanence de 0.0 représente une synapse potentielle non valide qui n'a pas progressé du tout vers l'état valide. Une valeur au-dessus du seuil (typiquement 0.2) représente une synapse qui a juste commencé à se connecter mais qui pourrait être très facilement

déconnectée. Une valeur de permanence élevée, comme 0.9, représente une synapse qui est connectée et qui ne peut pas être facilement déconnectée.

Le nombre de synapses valides sur les segments de dendrites proximales et distales d'une cellule HTM n'est pas figé. Il varie en fonction des motifs auxquels la cellule est exposée. C'est ainsi que le nombre de synapses valides sur les dendrites distales dépend par exemple de la structure temporelle des données. Si aucun motif temporel ne se répète sur les entrées de la région alors toutes les synapses sur les segments distaux auront des valeurs de permanence très faibles et très peu de synapses seront valides. A l'inverse si les données d'entrée comportent beaucoup de structures temporelles on trouvera alors de nombreuses synapses valides.

Sortie de la cellule
Une cellule HTM présente deux sorties binaires distinctes : 1) la cellule est active en raison d'entrées aval (feed-forward input) via la dendrite proximale et 2) la cellule est active en raison de connexions latérales via les segments de dendrites distales. Le premier cas est appelé « état actif » et le second « état prédictif ».

Dans le diagramme ci-dessus, les deux sorties sont représentées par les lignes sortant de corps carré de la cellule. La ligne de gauche est l'état actif aval (feed-forward active state) et celle de droite l'état prédictif.

Seul l'état actif aval (feed-forward active state) est connecté aux autres cellules de la région, garantissant ainsi que les prédictions s'appuient toujours sur l'entrée courante (plus le contexte). Nous ne voulons surtout pas faire de prédiction en s'appuyant sur des prédictions. Si nous le faisions, presque toutes les cellules de la région se retrouveraient en état prédictif après seulement quelques itérations.

La sortie de la région est un vecteur représentant l'état de toutes les cellules. Ce vecteur devient l'entrée de la prochaine région dans la hiérarchie s'il en existe une. Cette sortie est un OU booléen des états actifs et prédictifs. En les combinant la sortie de la région sera plus stable (évolution plus lente) que l'entrée. Cette stabilité est une importante propriété de l'inférence d'une région.

Suggestions de lecture

On nous demande souvent de recommander des lectures permettant d'en apprendre davantage sur les neurosciences. Le champ des neurosciences est si large qu'une introduction générale demande de puiser dans plusieurs sources. Les nouvelles découvertes sont publiées dans des journaux scientifiques qui sont à la fois ardus et difficiles d'accès si vous n'avez pas d'affiliation universitaire.

Voici deux livres disponibles qu'un lecteur motivé pourra consulter et qui couvrent les sujets de cette annexe.

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

Cet ouvrage est une bonne source pour tout ce qui concerne les dendrites. Le chapitre 16 traite des propriétés non linéaires des segments dendritiques utilisés dans les algorithmes d'apprentissage corticaux HTM. Il a été rédigé par Bartlett Mel qui a effectué la plupart des travaux dans ce domaine.

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

Cet ouvrage est une bonne introduction à tout ce qui touche au cortex cérébral. Plusieurs chapitres sont consacrés aux différents types de cellules et à leurs connexions. Vous pourrez ainsi vous faire une bonne idée des neurones corticaux et de leurs connexions même si cet ouvrage est trop ancien pour faire état des dernières connaissances sur les propriétés des dendrites.

Annexe B : comparaison des couches du cortex cérébral et d'une région HTM

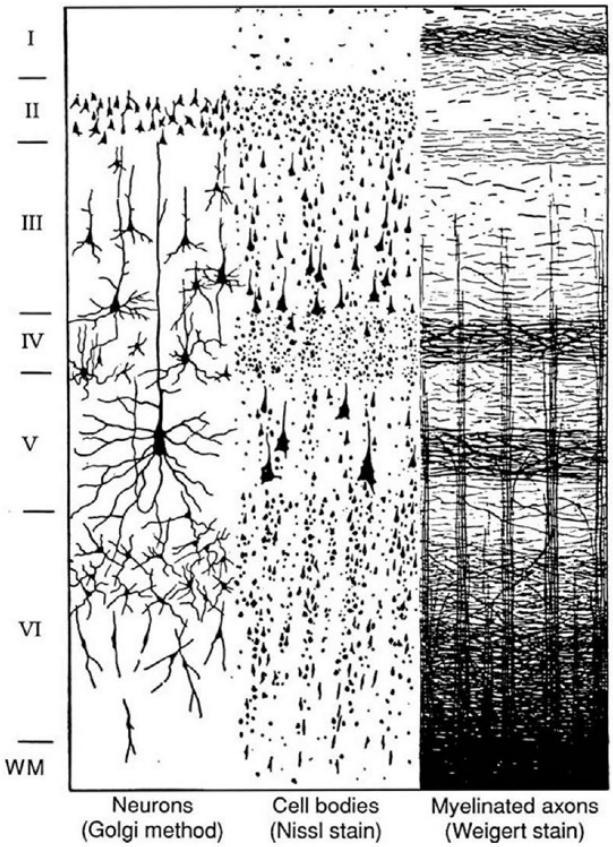
Cette annexe décrit la relation entre une région HTM et une région du cortex cérébral.

Plus précisément, l'annexe explique de quelle façon l'algorithme d'apprentissage cortical HTM avec ces colonnes et cellules est lié à l'architecture en couches et en colonnes du cortex cérébral. Nombreuses sont les personnes désorientées par le concept de « couches » du cortex cérébral et par leurs liens avec les couches HTM. Nous espérons que cette annexe permettra de lever cette confusion et donnera une meilleure vision de la biologie sous-jacente à l'algorithme d'apprentissage cortical HTM.

Circuiterie du cortex cérébral

Le cortex humain est une feuille de tissu neural d'environ 1000 cm^2 de surface et 2mm d'épaisseur. Pour visualiser cette feuille, songez à une serviette de table en tissu : c'est une bonne approximation de la surface et de l'épaisseur du cortex cérébral. Le cortex est divisé en une douzaine de régions fonctionnelles, certaines liées à la vision, d'autres à l'audition et d'autres encore au langage, etc… Vu au microscope, les caractéristiques physiques des différentes régions sont remarquablement similaires.

On retrouve dans chaque région du cortex plusieurs principes d'organisation récurrents.



Couches

On dit généralement du cortex cérébral qu'il possède six couches. Cinq de ces couches contiennent des cellules et la dernière est essentiellement faite de connexions. Ces couches ont été découvertes il y a plus d'un siècle grâce à l'arrivée des techniques de coloration microscopique. L'image ci-dessus (de Cajal) montre une mince tranche de cortex révélée avec trois techniques de coloration différentes. L'axe vertical de la coupe traverse l'épaisseur du cortex soit approximativement 2 mm. La partie gauche de l'image montre les 6 couches. La couche 1, au sommet, ne possède pas de cellule. L'acronyme « WM[3] » au bas de l'image indique le début de la matière blanche où les axones des cellules cheminent vers d'autres parties du cortex et du cerveau.

La partie droite de l'image utilise une coloration qui met en évidence les axones myélinisés (la myélinisation se traduit par une gaine lipidique qui recouvrent certains axones). Dans cette partie de l'image on observe deux des grands principes d'organisation du cortex, les couches et les colonnes. La plupart des axones se séparent en deux juste après avoir quitté le corps du neurone. Une branche progresse pratiquement à l'horizontal et l'autre à la vertical. La branche horizontale établit un grand nombre de connexions avec des cellules de la même couche ou

[3] NdT : de l'anglais « White Matter »

d'une couche proche rendant ainsi visible la division en couche par coloration. Gardez présent à l'esprit qu'il s'agit ici d'un dessin d'une tranche de cortex. La plupart des axones rentrent et sortent du plan de l'image et les axones sont donc en fait bien plus long qu'ils ne le paraissent sur l'image. On estime qu'il y a entre 2 et 4 km d'axones et de dendrites dans chaque millimètre cube de cortex.

La partie centrale de la figure montre une coloration des corps des neurones sans les dendrites ni les axones. Vous pouvez aussi constater que la taille et la densité des neurones varient selon les couches. Sur cette image on note à peine la structure en colonne. Vous avez aussi peut-être remarqué qu'il y a quelques neurones dans la couche 1. Leur nombre est si faible dans cette couche qu'on la désigne souvent sous le vocable de couche non-cellulaire. Les neuroscientifiques ont estimé qu'il y a environ 100 000 neurones par millimètre cube de cortex.

La partie gauche de l'image teinte le corps, les axones et les dendrites de quelques neurones. Vous constaterez que la taille des « arbres » dendritiques varie selon les cellules des différentes couches. On y voit aussi quelques « dendrites apicales » qui émergent des corps des cellules pour établir des connexions dans d'autres couches. La présence et la destination des dendrites apicales sont spécifiques à chaque couche.

En résumé, la structure en couche et en colonne du cortex devient évidente lorsque les tissus teintés sont observés au microscope.

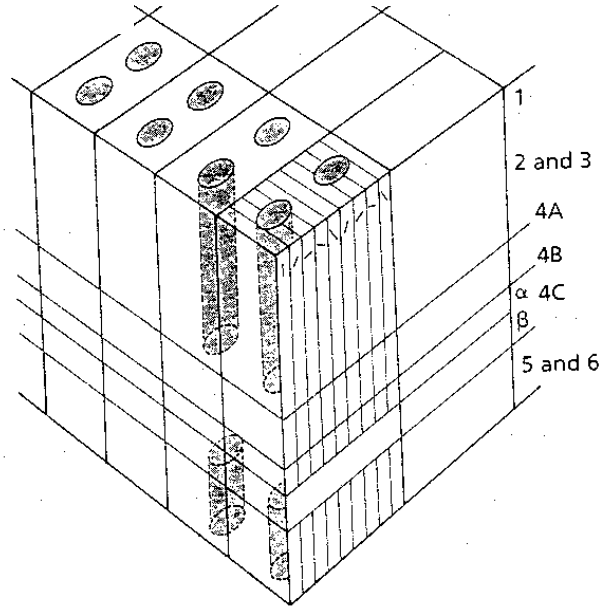
Variations des couches selon les régions

Il existe des variations dans l'épaisseur des couches selon les régions du cortex et des désaccords concernant le nombre de couches lui-même. Ces variations dépendent de l'animal étudié, de la région observée et de l'observateur. Par exemple, dans la figure ci-dessus, les couches 2 et 3 sont aisément discernables mais ce n'est généralement pas le cas. Certains chercheurs indiquent qu'ils ont été incapables de distinguer ces deux couches dans les régions qu'ils ont étudiées et c'est pour cette raison que les couches 2 et 3 sont souvent regroupées et baptisées « couche 2/3 ». A l'inverse d'autres scientifiques vont jusqu'à définir des sous-couches telles que 3A et 3B.

La couche 4 est la mieux définie dans les régions du cortex qui sont proches des organes sensoriels. Alors que chez certains animaux (tel l'homme et les singes) la couche 4 est parfaitement délimitée dans la première région de la vision chez d'autres elle est indiscernable. La couche 4 disparait dans les régions hiérarchiquement éloignées des organes sensoriels.

Colonnes
Les colonnes sont le second grand mode d'organisation du cortex. L'organisation colonnaire est visible sur les images teintées mais la preuve de l'existence des colonnes s'appuie surtout sur la façon dont les cellules répondent à différentes entrées.

Lorsque les scientifiques utilisent des sondes pour observer ce qui active les neurones ils constatent que les neurones qui sont verticalement alignés le long des différentes couches répondent à peu près à l'identique à une entrée donnée.



Ce schéma illustre quelques-unes des propriétés des réponses fournies par les cellules de V1, la première région en charge du traitement des informations en provenance de la rétine.

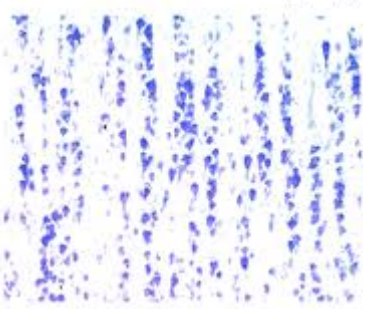
Une des premières découvertes a montré que la plupart des cellules de V1 répondent à des images de lignes et d'arêtes ayant une orientation donnée pour une zone donnée de la rétine. Les cellules alignées verticalement dans les colonnes répondent toutes à des arêtes ayant la même orientation. Si vous regardez attentivement, vous verrez que le schéma montre un ensemble de lignes courtes orientées différemment et disposées tout en haut de la coupe. Ces petites lignes indiquent à quelle orientation répondent les cellules placées à cet endroit. Les cellules qui sont verticalement alignées répondent toutes aux lignes orientées d'une certaine façon.

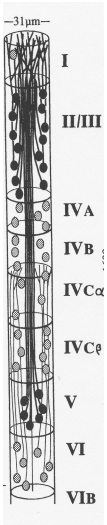
Il existe plusieurs autres propriétés attachées aux colonnes de V1 dont deux sont visibles sur le schéma. Il existe des « colonnes à dominante oculaire » où les cellules répondent à des combinaisons semblables d'influence de l'œil gauche et droit. Et il existe d'autres « poches » où les cellules sont essentiellement sensibles à la couleur. Les colonnes à dominante oculaire représentent les blocs les plus grands dans le schéma. Chacune d'elle intègre un ensemble de colonnes d'orientation. Les « poches » sont les ovales noirs.

La règle générale qui prévaut au sein du cortex semble indiquer que différentes propriétés de réponse se recouvrent l'une l'autre comme l'orientation et la dominance oculaire. En vous déplaçant horizontalement à la surface du cortex la combinaison des propriétés de réponse affichées par les cellules changent. Toutefois, les neurones alignés verticalement partagent le même jeu de propriétés de réponse. Cette règle s'applique dans les zones auditive, visuelle et somatosensorielle. Il existe encore un débat entre neuroscientifiques sur l'universalité de ce mode d'organisation sur la totalité du cortex mais il est certain qu'il apparait dans la plupart des zones du cortex sinon toutes.

Mini-colonnes

La plus petite structure colonnaire du cortex est la mini-colonne. Les mini-colonnes mesurent à peu près 30 µm de diamètre et comprennent de 80 à 100 neurones sur la traversée des 5 couches de neurones. Le cortex entier est composé de mini-colonnes. Vous pouvez vous les représenter comme des petits bouts de spaghettis serrés les uns contre les autres. Il existe des petits espaces avec très peu de cellules entre les mini-colonnes ce qui les rend parfois visibles sur des préparations microscopiques teintées.





Sur la gauche de l'illustration figure une image teintée montrant des corps de neurones dans une tranche de cortex. La structure verticale des mini-colonnes saute aux yeux sur cette image. Sur la droite figure le dessin conceptualisé d'une mini-colonne (tiré de Peters and Yilmez). Elle est en réalité plus fine que cela. Notez que

dans chaque couche de la colonne figurent plusieurs neurones. Tous les neurones d'une mini-colonne répondront de la même manière à une entrée donnée. Par exemple, dans le dessin de la section V1 montrée précédemment, une mini-colonne contiendra des cellules qui répondent à des lignes à l'orientation particulière avec une préférence oculaire dominante. Les cellules d'une colonne adjacente pourront répondre à une orientation légèrement différente ou une préférence oculaire dominante différente.

Les neurones inhibiteurs jouent un rôle essentiel dans la définition des mini-colonnes. Ils ne sont visibles ni sur l'image ni sur le dessin mais les neurones inhibiteurs projettent leurs axones entre les mini-colonnes les séparant ainsi physiquement. On pense aussi que les neurones inhibiteurs contribuent à forcer toutes les cellules de la mini-colonne à répondre à l'identique à une entrée donnée.

La mini-colonne est le modèle utilisé pour les colonnes de l'algorithme d'apprentissage cortical HTM.

Une exception aux réponses par colonne

Il existe une exception aux réponses par colonne qui s'avère pertinente pour l'algorithme d'apprentissage cortical HTM. Habituellement les chercheurs identifient à quoi répond une cellule en exposant des animaux d'expérimentation a de stimuli simples. Par exemple on peut lui montrer une simple ligne dans une partie de son champ visuel pour déterminer quelle est la réponse des cellules de la région V1. En utilisant des entrées simples, les chercheurs ont découvert qu'une entrée donnée déclenche toujours les mêmes cellules. Toutefois si cette même entrée simple est plongée dans une vidéo d'une scène naturelle les cellules deviennent plus sélectives. Une cellule qui répond de façon systématique à une ligne verticale isolée ne se déclenchera pas toujours si cette même ligne se retrouve dans une image complexe et en mouvement.

Dans l'algorithme d'apprentissage cortical HTM, toutes les cellules HTM d'une colonne partage les mêmes propriétés de réponse aval (feed-forward response) mais dans une séquence temporelle apprise, seule une des cellules de la colonne HTM devient active. Ce mécanisme est le moyen par lequel on représente des séquences d'ordre variable et il est analogue à la propriété des neurones décrite précédemment. Une entrée simple sans contexte provoquera l'activation de toutes les cellules d'une colonne. La même entrée dans une séquence apprise activera uniquement une cellule.

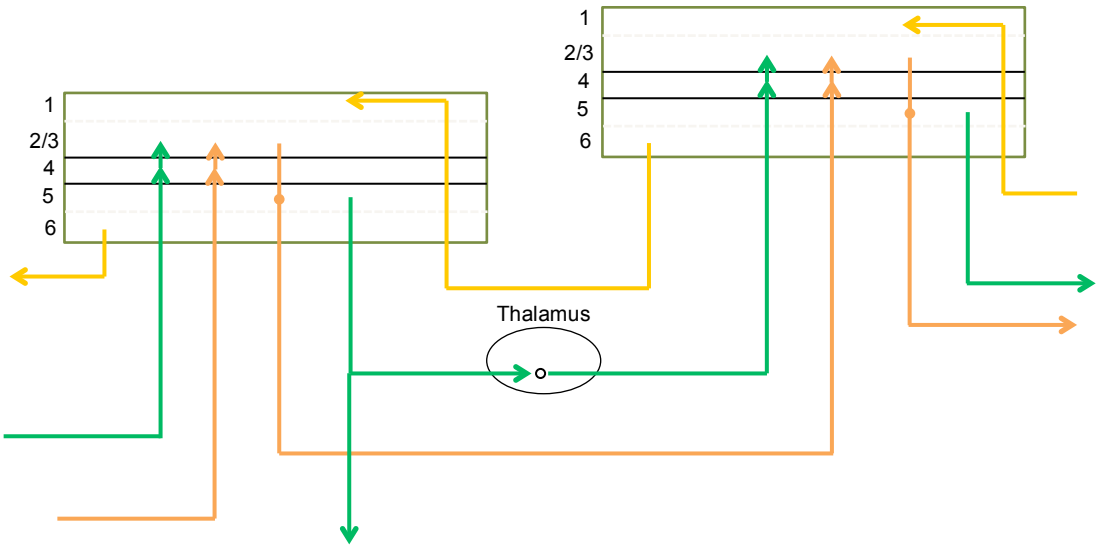
Nous ne sommes pas en train de suggérer qu'un seul neurone d'une mini-colonne sera actif à un moment donné. L'algorithme d'apprentissage cortical HTM suggère en fait que dans une colonne, tous les neurones d'une couche seront actifs sur une entrée inattendue et uniquement un sous-ensemble pour une entrée anticipée.

Pourquoi y a-t-il des couches et des colonnes?

Personne ne sait véritablement pourquoi le cortex est fait de couches et de colonnes. La théorie HTM propose toutefois une explication. L'algorithme d'apprentissage cortical HTM montre qu'une couche de cellules organisée en colonnes s'avère être une mémoire de transitions d'états d'ordre variable de très haute capacité. Dit plus simplement, une couche de cellules organisées de cette manière peut apprendre beaucoup de séquences. Des colonnes de cellules qui partagent la même réponse aval (feed-forward response) constituent le mécanisme clé pour l'apprentissage efficace de transitions d'ordre variable.

Cette hypothèse explique pourquoi les colonnes sont nécessaires mais qu'en est-il des couches ? Si une seule couche corticale peut apprendre des séquences et établir des prédictions pourquoi en voyons-nous 5 dans le cortex cérébral ?

Nous conjecturons que les différentes couches observées dans le cortex apprennent toutes des séquences en s'appuyant sur le même mécanisme mais que les séquences apprises à chaque niveau sont utilisées différemment. Il y encore beaucoup de choses que nous ne comprenons pas à ce sujet mais nous pouvons néanmoins décrire ici l'idée générale. Avant cela, il est utile de décrire à quoi se connectent les neurones de chaque couche.



Le diagramme ci-dessus illustre deux régions du cortex et leurs connexions. Ces connexions peuvent être observées dans tout le cortex lorsque deux régions se projettent l'une vers l'autre. La boite de gauche représente une région corticale hiérarchiquement plus basse que la région (boite) de droite, de sorte que les informations aval (feed-forward information) circulent de gauche à droite sur le diagramme. Les flèches descendantes se projettent vers d'autres régions du cortex. La rétro information (information amont ou feedback information) va de droite à

gauche. Chaque région est divisée en couches. Les couches 2 et 3 sont ici amalgamées sous l'appellation couche 2/3.

Les lignes colorées représentent la sortie des neurones dans les différentes couches. Ce sont des regroupements d'axones émanant des neurones de la couche. Souvenez-vous que les axones de divisent rapidement en deux. Une branche qui s'étend à l'horizontal dans la région et principalement dans la même couche. Ainsi toutes les cellules d'une couche sont fortement connectées. Les neurones et les connexions horizontales ne sont pas montrés sur le diagramme.

Il existe deux chemins de communication aval, un direct dessiné en orange sur la figure et un indirect en vert. La couche 4 est la principale couche d'entrée aval et reçoit ses propres entrées des deux chemins de communication aval. La couche 4 se projette dans la couche 3.

La couche 3 est aussi l'origine du chemin aval direct. Ainsi le chemin aval direct est limité aux couches 3 et 4.

Quelques connexions aval passent outre la couche 4 et vont directement à la couche 3. Et, comme mentionné plus haut, la couche 4 disparait dans les régions éloignées des zones sensorielles. Dans ces régions, le chemin aval direct passe donc uniquement de la couche 3 d'une région à la couche 3 de la région suivante.

Le second chemin de communication aval (en vert) prend naissance en couche 5. Les cellules de la couche 3 établissent des connexions vers les cellules de la couche 5 avant de finir leur chemin dans la région suivante. Après être sorties de la feuille corticale, les axones de la couche 5 se séparent à nouveau en deux. Une branche se projette dans les zones subcorticales du cerveau impliquées dans la motricité. On pense que ces axones pilotent notre motricité (représentés ici avec une flèche orientée vers le bas). L'autre branche se projette vers une partie du cerveau appelée thalamus qui agit comme une barrière laissant passer l'information vers la prochaine région ou pas.

Pour finir, le chemin principal de rétro-information, montré en jaune, débute en couche 6 et se projette dans la couche 1. Les cellules des couches 2, 3 et 5 se connectent à la couche 1 via leur dendrite apicale (non représentée). La couche 6 reçoit ses entrées de la couche 5.

Cette description est un résumé sommaire de ce que nous savons aujourd'hui des connexions entre couches mais elle sera suffisante pour comprendre notre hypothèse sur l'existence de plusieurs couches d'apprentissage.

Hypothèses sur ce que font les couches

Dans nos hypothèses, nous proposons que les couches 3, 4 et 5 soient toutes des couches à réponse aval (feed-forward layers) et que toutes fassent de l'apprentissage de séquence. La couche 4 apprend les séquences de premier ordre. La couche 3 apprend les séquences d'ordre variable. Et la couche 5 apprend les séquences d'ordre variable y compris le séquencement temporel. Examinons chacune d'elles plus en détail.

Couche 4
Il est facile d'apprendre des séquences de premier ordre en utilisant l'algorithme d'apprentissage cortical HTM. Si on ne force pas les cellules d'une colonne à s'inhiber l'une l'autre, c'est-à-dire que les cellules ne se différencient pas selon les entrées précédemment fournies alors elles apprennent des séquences de premier ordre. Dans le cortex cérébral on parviendrait probablement au même résultat en supprimant l'effet inhibiteur entre les cellules d'une même colonne. Dans notre modèle informatique de l'algorithme d'apprentissage cortical HTM, en disposant une seule cellule par colonne on arrive à un résultat similaire.

Des séquences de premier ordre c'est ce dont nous avons besoin pour former les représentations invariantes des transformations spatiales d'une entrée. Dans la vision par exemple, les translations en x-y, le changement d'échelle et la rotation sont des transformations spatiales. Quand une région HTM effectue son apprentissage sur des objets en mouvement, elle apprend les équivalences entre les différentes représentations spatiales. Les cellules HTM qui ressortent de ce processus se comportent comme ce qu'on appelle les « cellules complexes » du cortex. Les cellules HTM demeureront actives (en état prédictif) sur un ensemble de transformations spatiales.

Chez Numenta nous avons procédé à des expérimentations dans le domaine de la vision qui confirme ce mode de fonctionnement et qu'une invariance spatiale est bien atteinte à chaque niveau. Les détails de cette expérimentation dépassent le cadre de ce document.

L'apprentissage de séquences de premier ordre dans la couche 4 est cohérent avec le fait qu'on y trouve des cellules complexes et aussi avec sa disparation dans certaines régions hautes du cortex. En montant dans la hiérarchie il arrive un point au-delà duquel il n'est plus possible d'apprendre davantage d'invariances spatiales car les représentations à ce niveau y seront déjà insensibles.

Couche 3
La couche 3 la plus proche de l'algorithme d'apprentissage cortical décrit au Chapitre 2. Elle apprend des séquences d'ordre variable et établit des prédictions qui sont plus stables dans le temps que ses entrées. La couche 3 se projette toujours dans la prochaine région de la hiérarchie et conduit donc à une stabilité temporelle accrue dans la hiérarchie. La mémoire de séquences d'ordre variable nous amène aux neurones dits « cellules complexes directionnelles » (directionally-tuned complex cells) qu'on observe dans la couche 3. Les cellules complexes

directionnelles se différencient par leur capacité à distinguer un contexte temporel tel qu'une ligne se déplaçant vers la gauche d'une ligne se déplaçant vers la droite.

Couche 5

La dernière couche à flux aval est la couche 5. Nous conjecturons que la couche est similaire à la couche 3 à trois différences près. La première est que la couche 5 ajoute le concept de timing. La couche 3 prédit ce qui va survenir mais elle ne dit pas quand. Or de nombreuses taches requièrent une appréhension du timing comme la reconnaissance de mots prononcés où le timing relatif entre phonèmes est important. Le comportement moteur est un autre exemple : la coordination dans le temps de l'activation des muscles est essentielle. Nous conjecturons donc que la couche 5 prédit le prochain état après le laps de temps attendu. Plusieurs détails biologiques soutiennent cette hypothèse. La première c'est que la couche 5 est la couche de commande de la motricité du cortex. Ensuite la couche 5 reçoit ses entrées de la couche 1 qui prend naissance dans une région du thalamus (non représenté sur le diagramme). Nous émettons l'hypothèse que c'est par le biais d'une entrée thalamique dans la couche 1 que le temps est encodé et distribué à de nombreuses cellules.

La seconde différence entre la couche 3 et la couche 5, c'est que nous attendons de la première qu'elle fasse des prédictions aussi loin que possible dans le futur, apportant ainsi une stabilité temporelle. C'est ce que fait l'algorithme d'apprentissage cortical HTM décrit au chapitre 2. A l'opposé, nous souhaitons que la couche 5 prédise simplement le prochain évènement (qui surviendra à un instant donné). Nous n'avons pas modélisé cette différence entre les couches 3 et 5 mais cela se ferait naturellement si chaque transition attendue était associé à un temps.

La troisième différence entre les couches 3 et 5 peut être observée sur le diagramme. La sortie de la couche 5 se projette toujours dans les centres moteurs sous-corticaux et le chemin aval est contrôlé par le thalamus. La sortie de la couche 5 est parfois transférée à la prochaine région et parfois bloquée. Nous-mêmes et d'autres émettons l'hypothèse que ce contrôle est lié au phénomène d'attention cachée (covert attention) (l'attention cachée décrit le fait de porter attention à une information sans que cela se traduise par un comportement moteur).

En résumé, la couche 5 combine les capacités de timing spécifique, d'attention et de comportement moteur. De nombreux mystères entourent encore l'articulation entre ces trois capacités. Le point que nous souhaitons mettre en avant c'est que l'algorithme d'apprentissage cortical HTM peut facilement intégrer la notion de timing spécifique et permet aussi d'expliquer le rôle de la couche 5 du cortex.

Couche 2 et couche 6

La couche 6 est la couche d'origine des axones qui renvoie de l'information dans les couches basses. On en sait par contre peu sur la couche 2. Comme mentionné précédemment, l'existence même de la couche 2 comme distincte de la couche 3 est même l'objet de débats. Nous n'en dirons pas davantage sur cette question si ce

n'est de souligner que les couches 2 et 6, comme toutes les autres couches, montrent les propriétés de connexions horizontales massives et de réponse en colonne. Pour cette raison nous émettons l'hypothèse qu'elles déroulent aussi une variante de l'algorithme d'apprentissage cortical HTM.

À quoi correspond une région HTM dans le cortex ?
Nous avons implémenté l'algorithme d'apprentissage cortical HTM selon deux variantes, l'une avec plusieurs cellules par colonne pour la mémoire d'ordre variable et l'autre avec une seule cellule par colonne pour la mémoire de premier ordre. Nous pensons que ces deux variantes correspondent respectivement à la couche 3 et à la couche 4 du cortex. Nous n'avons pas essayé de combiner ces deux variantes en une seule région HTM.

Bien que l'algorithme d'apprentissage cortical HTM (à plusieurs cellules par colonnes) soit plus proche de la couche 3 du cortex, nous disposons d'une flexibilité dans nos modèles que le cerveau n'a pas. Ainsi nous pouvons créer des couches cellulaires hybrides qui ne correspondent à aucune des couches du cortex. Par exemple dans notre modèle nous savons dans quel ordre les synapses sont formés sur les segments dendritiques. Nous pouvons utiliser cette information pour extraire ce qu'il va se passer juste après parmi l'ensemble des prédictions de tout ce qu'il peut se passer dans le futur. Nous pouvons probablement ajouter aussi la notion de timing de la même façon. Il devrait ainsi être possible de créer une région HTM composée d'une seule couche combinant les propriétés des couches 3 et 5 du cortex.

Résumé

L'algorithme d'apprentissage cortical HTM représente ce que nous pensons être une brique de base de l'organisation des neurones dans le cortex. Il montre comment une couche de neurones connectés horizontalement apprend des séquences d'information représentées de façon parcimonieuses et distribuées. Des variantes de l'algorithme d'apprentissage cortical HTM sont utilisées dans les différentes couches du cortex dans des buts différents mais liées entre eux.

Nous proposons que les entrées aval (feed-forward input) d'une région du cortex cérébral que ce soit dans la couche 4 ou la couche 3 se projettent de façon prédominante sur les dendrites proximales, qui avec l'aide des cellules inhibitrices, créent une SDR de l'entrée. Nous conjecturons que les cellules des couches 2, 3, 4, 5 et 6 partagent cette même SDR en obligeant toutes les cellules d'une colonne traversant l'ensemble des couches à répondre à une même entrée aval (feed-forward input).

Nous suggérons que les cellules de la couche 4, lorsqu'elles sont présentes, utilise l'algorithme d'apprentissage cortical HTM pour apprendre des transitions

temporelles de premier ordre et produire des représentations invariantes des transformations spatiales. Les cellules de la couche 3 utilisent l'algorithme d'apprentissage cortical HTM pour apprendre des transitions temporelles d'ordre variable et former des représentations stables transmises aux couches supérieures dans la hiérarchie du cortex. Les cellules de la couche 5 apprennent les transitions d'ordre variable avec timing. Nous n'avons pas de proposition particulières ni pour la couche 2 ni pour la couche 6. Toutefois, en raison de la connectivité horizontale typique de ces couches il est très probable qu'elles constituent elles aussi une mémoire de séquence.

Glossaire

Notes: les définitions ci-dessous indiquent comment les termes sont utilisés dans ce document et pour certains dans un sens différent de l'usage général. Les termes commençant par une lettre majuscule dans les définitions font référence à d'autres termes du glossaire.[4]

Etat actif (Active State)	un état dans lequel les Cellules sont actives en raison d'une entrée Aval.
Ascendant (Bottom-Up)	synonyme de Aval
Cellules (Cells)	L'équivalent HTM d'un Neurone *Les Cellules sont organisées en colonne dans les régions HTM.*
Activité coïncidente (Coincident Activity)	deux Cellules ou plus sont actives au même moment
Colonne (Column)	un groupe d'une ou plusieurs Cellules qui fonctionnent comme une seule entité dans une région HTM *Les Cellules d'une colonne donnée représentent la même entrée aval mais dans des contextes différents.*
Segment de dendrites (Dendrite Segment)	Une unité d'intégration de Synapses associées à des Cellules et des Colonnes. *Les MTHs ont 2 types de segments de dendrites. L'un est associé aux connexions latérales vers une cellule. Lorsque le nombre synapses actives sur le segment de dendrites dépasse un certain seuil, les cellules associées entrent dans l'état prédictif. L'autre est associé avec des connexions aval dans une colonne. Le nombre de synpases actives s'additionne pour générer l'activation Aval d'une colonne.*
Densité souhaitée (Desired Density)	Pourcentage souhaité de Colonnes qui sont actives en raison d'une entrée Aval vers une région *Le pourcentage s'applique uniquement dans un rayon qui varie en fonction de l'étalement (fan-out) des entrées aval. Il est « désiré » car le pourcentage peut varier en fonction d'une entrée particulière.*

[4] NdT : nous avons volontairement laissé dans le glossaire les termes anglais originaux

Aval ou Avant (Feed-Forward)	se déplaçant dans une direction qui s'éloigne d'une entrée ou bien d'un Niveau bas vers un Niveau plus haut de la Hiérarchie (aussi appelé Ascendant)
Amont ou Arrière (Feedback)	se déplaçant dans une direction qui retourne vers une entrée ou bien d'un Niveau haut vers un niveau plus bas de la Hiérarchie (aussi appelé Descendant)
Prédiction de premier ordre (First Order Prediction)	une prédiction basée sur l'entrée courante et non sur les précédents – à comparer à une Prédiction d'ordre variable
Mémoire temporelle hiérarchique - HTM (Hierarchical Temporal Memory - HTM)	Une technologie qui reproduit quelques-unes des fonctions structurelles et algorithmiques du cortex.
Hiérarchie (Hierarchy)	Un réseau d'éléments connectés où les connexions sont aval ou amont
Algorithme d'apprentissage cortical HTM (HTM Cortical Learning Algorithms)	L'ensemble des fonctions du Concentrateur spatial, du Concentrateur temporel, d'apprentissage et d'oubli d'une région HTM. Aussi appelé Algorithmes d'apprentissage HTM.
Réseau HTM (HTM Network)	Une Hiérarchie de Régions HTM
Région HTM (HTM Region)	L'unité principale de mémoire et de prédiction d'une HTM *Une région HTM est composée d'une couche de cellules fortement interconnectées disposées en colonnes. Aujourd'hui une région HTM n'a qu'une couche de cellules alors que le cortex (et plus tard une région HTM) en possède plusieurs. Lorsqu'on se réfère à une région du point de vue de sa position dans une hiérarchie on parle de Niveau.*
Inférence (Inference)	Reconnaitre un motif spatial et temporel en entrée comme étant similaire à un des motifs précédemment appris
Rayon d'inhibition (Inhibition Radius)	définit la zone autour d'une colonne sur laquelle elle a un effet d'inhibition.
Connexions latérales (Lateral Connections)	connexions entre Cellules d'une même Région
Niveau (Level)	Une région HTM donnée dans le contexte d'une Hiérarchie

Neurone (Neuron)	Une Cellule de traitement de l'information du cerveau *Dans ce document nous utilisons le mot Neurone pour faire spécifiquement référence aux cellules biologiques du cerveau et nous utilisons le mot Cellules lorsque nous parlons de l'unité de traitement HTM.*
Permanence (Permanence)	Une valeur scalaire qui indique l'état de connexion d'une Synapse potentielle. *Une valeur de permanence inférieure à un certain seuil indique une synapse non formée. Une valeur de permanence supérieure à un certain seuil indique une synapse valide. L'apprentissage dans une région HTM se fait par l'ajustement des valeurs de permanence des Synapses potentielles..*
Synapse potentielle (Potential Synapse)	Le sous-ensemble de toutes les Cellules qui peuvent potentiellement former des Synapses avec un Segment de dendrites donné. *À un instant donné seul un sous-ensemble des synapses sont valides en fonction de leur valeur de permanence.*
Prédiction (Prediction)	Activer des Cellules à l'état prédictif qui deviendront vraisemblablement Actives dans un futur proche par le biais d'une entrée Aval. *Une région HTM prédit souvent plusieurs futures entrées simultanément.*
Champ de réception (Receptive Field)	L'ensemble des entrées auxquelles une Colonne ou une Cellule est connectée. *Si l'entrée d'une région HTM est organisée sous forme d'un tableau de bits à 2 dimensions alors le champ de réception peut s'exprimer sous la forme d'un rayon sur l'espace des entrées.*
Capteur (Sensor)	Une source d'entrées pour un Réseau HTM
Représentation distribuée parcimonieuse (Sparse Distributed Representation)	Représentation composée de nombreux bits dont un faible pourcentage sont actifs et où aucun bit n'est suffisant à lui seul pour porter un sens.

Concentrateur spatial (Spatial Pooling)	Le processus qui amène à une SDR d'une entrée *L'une des propriétés du concentrateur spatial tient au fait que les motifs en entrée qui se recoupent sont projetés sur la même représentation parcimonieuse distribuée.*
Sous -échantillonnage (Sub-Sampling)	Reconnaitre un grand motif distribué en mettant uniquement en correspondance en petit sous-ensemble des bits actifs du motif.
Synapse (Synapse)	Connexions entre Cellules formées pendant l'apprentissage
Concentrateur temporel (Temporal Pooling)	Le processus qui amène à la formation d'une représentation d'une séquence de motifs d'entrée qui est temporellement plus table que l'entrée elle-même.
Descendant(es) (Top-Down)	Synonyme de Amont ou Arrière
Prédiction d'ordre variable (Variable Order Prediction)	Une prédiction basée sur un nombre variable de contextes antérieurs – à comparer à une Prédiction de premier ordre *Elle est appelée « variable » car la mémoire permettant de maintenir le contexte antérieur est allouée selon les besoins. Ainsi un système de mémoire qui implémente la prédiction d'ordre variable peut utiliser des contextes remontant loin dans le temps sans demander des quantités exponentielles de mémoire.*