

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur : ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite de ce travail expose à des poursuites pénales.

Contact : portail-publi@ut-capitole.fr

LIENS

Code la Propriété Intellectuelle – Articles L. 122-4 et L. 335-1 à L. 335-10

Loi n°92-597 du 1^{er} juillet 1992, publiée au *Journal Officiel* du 2 juillet 1992

<http://www.cfcopies.com/V2/leg/leg-droi.php>

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>



THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 1 Capitole (UT1 Capitole)*

Présentée et soutenue le *12/12/2014* par :

ANGELA BOVO

Apprentissage automatique pour l'assistance au suivi d'étudiants en ligne : approches classique et bio-inspirée

JURY

MATHIEU MURATET	Maître de conférences
DAVID PANZOLI	Maître de conférences
PIERRE DE LOOR	Professeur des universités
STÉPHANIE JEAN-DAUBIAS	Professeure des universités
STÉPHANE SANCHEZ	Maître de conférences
YVES DUTHEN	Professeur des universités

École doctorale et spécialité :

MITT : Image, Information, Hypermedia

Unité de Recherche :

Institut de recherche en informatique de Toulouse - UMR 5505

Directeur de Thèse :

Yves Duthen

Co-encadrant :

Stéphane Sanchez

Rapporteurs :

Pierre de Loor et Stéphanie Jean-Daubias

Remerciements

Le chemin qui m'a menée vers cette thèse remonte très loin. Sans aller jusqu'à l'époque où ma grand-mère m'apprenait à compter avec des piles de dix tickets de bus, on peut peut-être s'arrêter au moment où ma mère m'a abonnée à Science et Vie junior. Dans ce magazine formidable, j'ai tout à la fois découvert un intérêt pour les sciences et une fascination pour le monde de la recherche. Sans cette plongée mensuelle dans un monde différent, jamais la littérature que j'étais n'aurait bifurqué au dernier moment vers une filière scientifique, et l'attrait de l'astrophysique ne m'aurait peut-être pas amenée à Supaéro.

Un peu plus près de moi, il y a aussi tous ces thésards qui ont commencé avant moi, que j'ai eu la chance de côtoyer et qui ont contribué à me montrer qu'une thèse était quelque chose d'atteignable. Le premier fut Glop, mythique ancien du Simu dans ma jeunesse et mon premier MJ de sans règle, mais il fut suivi de beaucoup d'autres, notamment mon grand-parrain bien-aimé Cyril, confident et dieu de l'impro à ses heures perdues, ma co-bureau de stage Stéphanie Roussel, et toute la bande du DCSD, pour ceux qui avaient déjà commencé à l'époque : Stéphane, Jack, Mario, Thibault, Caro, Julien, Julie, Alex, Patrice, et j'en oublie sûrement.

Mais ce qui m'en a donné vraiment envie, je le dois à Éric Bianchi, mon tuteur de stage à RodanoTech. En plus d'avoir bien encadré cette stagiaire qui lui tombait sur les bras dans un moment un peu difficile pour la société et d'avoir gentiment proposé de me covoiturier le soir, il s'est intéressé à mon cursus. En apprenant que je pensais me spécialiser dans l'intelligence artificielle en rejoignant le M2R associé, il m'a parlé de ce livre qu'il avait lu récemment, On Intelligence, d'un certain Jeff Hawkins, qui proposait une description du fonctionnement du cerveau. Sa publicité a bien fonctionné puisque je me suis procuré le livre peu de temps après la fin de mon stage. Comme lui, je l'ai dévoré et j'ai été passionnée. J'avais trouvé mon sujet de thèse.

Restait encore à y accéder. Supaéro et l'université Paul Sabatier s'accordent pour proposer un parcours de M2R simplifié pour les élèves de Supaéro, ce que je trouve vraiment formidable. J'ai donc suivi ce master et j'ai effectué mon stage à l'ONERA. Après ce stade, les choses se sont toutefois compliquées pour moi. En effet, le seul permanent de l'ONERA qui connaissait ces thématiques prenait sa retraite dans moins de trois ans, et n'aurait donc pas pu m'encadrer jusqu'au bout. Côté Supaéro, le seul encadrant possible m'a répondu qu'il avait déjà trop de thésards en ce moment. La seule bourse à la disposition du CerCo à ce moment-là était attribuée

par l'entreprise SpikeNet, qui a retenu un autre candidat.

C'est Nicolas Lassabe qui a débloqué la situation. Quand je lui ai parlé de mes recherches, il m'a orienté vers son ancien directeur de thèse, Yves Duthen, que j'ai donc rencontré. Yves m'en a mis plein la vue en me parlant des recherches de l'équipe et a accepté de prendre en charge la recherche de financement pour mon sujet. Je le remercie pour cette tâche ingrate, mais surtout pour m'avoir fait confiance. Bien sûr, ce n'était pas encore gagné. Plusieurs dossiers qu'il a montés ont essuyé des refus, et de mon côté, j'ai dû prendre mon mal en patience. Cette situation a duré six bons mois avant qu'Yves ne me présente à une entreprise de sa connaissance, Andil.

Lors de notre rencontre, j'ai été impressionnée par les réalisations d'Andil et plus encore par leur volonté de collaborer avec le monde de la recherche. Il s'agit à mon avis d'un positionnement à la fois audacieux, permettant de rester en pointe des technologies, et courageux, de par la prise de risque associée à l'essai de technologies non encore mures, surtout pour une petite entreprise. Nous avons fait une première réunion de brainstorming avec Stéphane Sanchez visant à déterminer l'adéquation entre les problématiques d'Andil, qui cherchait à créer un tuteur virtuel intelligent capable de guider les apprenants en ligne, et les sujets qui m'intéressaient. Cette adéquation nous ayant paru bonne, nous sommes ressortis de la réunion décidés à travailler ensemble. Nous avons alors rédigé un sujet de thèse avec Yves et Stéphane, puis monté un dossier CIFRE, qui a fini par être validé par l'ANRT après plusieurs mois d'attente. L'aventure pouvait enfin commencer.

Durant le cours de ma thèse, j'ai eu l'occasion de travailler avec bon nombre de personnes. Tout d'abord mon directeur de thèse, Yves, qui a toujours apporté à nos réflexions un grand recul, né de sa curiosité infatigable pour toutes les technologies nouvelles et les évolutions de la société. Je souhaite le remercier pour son encadrement et pour sa disponibilité malgré son emploi du temps toujours très rempli.

Stéphane, mon co-encadrant, a supporté pendant tout ce temps une thésarde particulièrement têtue qui avait un peu trop tendance à considérer les publications de Numenta comme une vérité suprême. . . Merci à lui pour ses conseils, pour son écoute patiente et ses suggestions quand rien ne marchait.

Olivier Héguy a été mon encadrant côté Andil. Il s'est toujours montré extrêmement disponible, patient et pédagogique, et la réputation de son Google-fu n'est plus à faire. Je le remercie de m'avoir initiée aux technologies employées par Andil et pour ses suggestions toujours pertinentes nourries par sa longue expérience de programmeur chevronné.

Alexandre Lesault et Nicolas Esplan, les dirigeants d'Andil, ont contribué à l'encadrement de la thèse en participant aux décisions de ses grandes orientations. Mais bien au-delà, je les remercie du fond du cœur pour la confiance qu'ils m'ont accordée et le risque qu'ils ont pris en choisissant de financer ma thèse, ainsi que pour la liberté qu'ils m'ont laissée dans mes travaux. Je crains que les résultats obtenus n'aient pas été à la hauteur de leurs espérances et j'en suis véritablement contrite.

Toujours à Andil, je voudrais remercier toute l'équipe pour le temps passé ensemble et la bonne ambiance. Merci à Arnaud pour l'aide linuxienne, le remplacement de mon

écran, et les discussions libristes ; à Alex pour les déboguages et pour les crêpes ; à Benoît pour la mise en place du blog et les chocolats ; à Jérôme pour le graphisme du blog et pour les discussions écolo-alternatives ; à Audrey pour m'avoir fait découvrir Fraish ; et à tous les autres, Laurie, Catarina, les gentils stagiaires, et les voisins de Juriscampus.

À l'UT1 aussi, j'ai été entourée de nombreux collègues. Dans la salle ME310, j'ai rejoint une fine équipe à mon arrivée : John et Paul, les deux larrons, toujours à mijoter les quatre cent coups et m'obligeant à verrouiller systématiquement ma session, Rémy, qui a gentiment supporté nos trolls sur Apple, Thomas, Daouda et Joseph. Par la suite nous ont rejoint encore bien d'autres gais lurons : Thai, Cesar, Nesrine, Nejma, Arnaud, Patrick, Diane, Bob, Maroussia, Maxime, qui a lancé la mode des jeux du midi, Meriam, Eunat, Jean, Caroline, Dennis, Martin. . . Et n'oublions pas bien sûr les permanents et les post-doc qui ont fait vivre notre petit groupe de l'UFR d'informatique d'UT1 : Fred, Hervé, Sylvain, Alain, Ronan, David, Nathalie, Chantal, Thomas, Nicolas, Benoît, Christophe, Laurent, Sylvie, Frank, Michèle et les autres que j'oublie sûrement. . . Merci à eux tous de leur compagnie, mais plus particulièrement à Rémy, qui est devenu un confrère de jeux de société et de jeux de rôle et avec qui j'ai toujours pu avoir des discussions particulièrement intéressantes, par exemple sur le transhumanisme.

Mes passages réguliers à l'IRIT m'ont permis de faire la connaissance notamment de Jean-Pierre Jessel, Louis-Pierre, Tom, et de toute l'équipe d'organisation de DocToMe, que je salue ici, ainsi que tout le personnel administratif côté MITT (Martine Labruyère), UT1 (Virginie Mangion) et IRIT, qui a toujours été extrêmement gentil et disponible.

Après le début et le milieu, toute thèse a une fin. Je souhaite remercier Stéphanie Jean-Daubias et Pierre de Loor d'avoir accepté de relire ma thèse et de la commenter. Alors que je doutais de sa qualité, leurs encouragements m'ont fait chaud au cœur, et toutes leurs remarques étaient extrêmement pertinentes.

Je remercie également Mathieu Muratet d'avoir accepté non seulement de participer, mais de présider le jury de ma thèse, et David Panzoli pour la qualité de sa relecture orthographique et scientifique ainsi que sa participation au jury.

Lors de ma soutenance, j'ai particulièrement apprécié la qualité de l'écoute et la bienveillance de tout le jury, et j'ai trouvé la discussion qui a suivi ma présentation très enrichissante, ce dont je souhaite également remercier tous les participants audit jury.

Je suis persuadée qu'il est très important de rester bien entouré et actif durant une thèse pour avoir de quoi se changer les idées. J'ai la chance d'avoir été plutôt bien servie dans ce domaine. Alors merci tout d'abord aux choristes de Il était une voix et à notre chef Mariande pour leur inextinguible bonne humeur, ainsi qu'aux professeurs bénévoles de Pasorock pour leur engagement à transmettre leur savoir, et aux membres de cette association avec lesquels nous avons passé de nombreuses soirées dansantes.

Merci également à Christophe Garion pour m'avoir proposé de co-encadrer des TP de Java à Supaéro. J'ai eu grand plaisir à retourner à l'école et à contribuer à transmettre le savoir que j'y avais appris, à peine quelques années plus tôt. À Virginie Wiels et Julien Brunel, qui ont été mes binômes lors de ces TP et avec qui j'ai eu grand plaisir à discuter, merci de m'avoir pleutresquement laissé vous refiler les erreurs de configuration Eclipse ou de SVN!

Simu un jour, Simu toujours? Nous avons fini par quitter le sous-sol de la R2, poussés à la fois par l'âge et par les pelleteuses, mais les liens que nous y avons créés durent toujours, et j'espère que ça sera encore longtemps le cas. Merci à tous mes amis, que je n'essaierai pas d'énumérer, toujours disponibles pour de la bonne bouffe et des activités ludiques. Certains ont quitté Toulouse, mais je ne vous oublie pas et j'espère toujours que vous reviendrez dans le coin. Quant aux bizuths qui tiennent actuellement le club, merci de supporter encore les vieux et leurs radoterics et d'avoir repris le flambeau avec honneur.

Ma famille m'a toujours soutenue en m'encourageant à poursuivre mes études, à choisir la voie qui m'intéressait et en me soutenant moralement et financièrement. Le prix à payer était de me voir m'éloigner à Toulouse, puis m'y installer, loin de vous. Mais vous n'êtes pas loin de mon cœur, et promis, je rentrerai plus souvent dès qu'il y aura un Hyperloop. J'ai parlé de famille, mais je dois préciser que mes remerciements y incluent également ma jolie-famille et la famille de mon parrain, que j'embrasse également très fort. Papa, Maman, je serai heureuse de vous voir à mes côtés le jour de la soutenance, j'espère qu'elle sera moins stressante que celle de Pascal. Mel, je sais que tu aurais voulu venir et que tu bosses dur, t'inquiète.

Le lecteur attentif aura remarqué une grande absence dans ce texte. J'ai préféré éviter les répétitions en ne mentionnant Pascal qu'une seule fois, et ce sera donc ici. En effet, j'aurais pu le citer dans à peu près tous les autres paragraphes. Pascal, tu as d'abord été mon parrain et mon modèle, me donnant la confiance en moi nécessaire pour faire le master et entamer ensuite une thèse à ta suite. C'est toi qui as contribué à me trouver mes encadrants, en parlant de moi à ton collègue Nicolas. Tu m'as ensuite soutenue financièrement durant toute la longue attente du démarrage de mon contrat. Tu as partagé mon enthousiasme pour mon sujet, et c'est avec toi que j'ai eu les discussions techniques les plus poussées. Ton grand talent de programmeur et ton esprit particulièrement vif m'ont plusieurs fois bien aidée. En plus de cette aide technique, tu as également été mon encadrant non officiel, avec qui j'ai toujours pu parler de mes difficultés pour recevoir des conseils toujours extrêmement pertinents, et tu as même relu bon nombre de textes et supporté des répétitions de présentation. Je ne sais vraiment pas comment j'aurais fait sans ton soutien et ta compréhension nés de ta propre expérience de la thèse, et qui continuent maintenant au moment de notre entrée dans le monde du travail. J'ai souvent été difficile à supporter et je regrette seulement de n'avoir pas pu t'apporter autant lorsque c'était ton tour. Au-delà de la thèse, je peux également te remercier en tant que choriste, danseur, ami... et famille, bientôt officiellement j'espère. Merci de partager ma vie.

Table des matières

1	Introduction	11
1.1	Avant-propos	11
1.2	Contexte du projet	11
1.3	Le numérique et l'éducation	12
1.3.1	Arrivée du numérique	12
1.3.2	Outils	13
1.3.3	Enjeux	15
1.3.4	La pédagogie à l'épreuve du numérique	16
1.4	L'apprentissage en ligne	18
1.4.1	Terminologie	18
1.4.2	Avantages et inconvénients	19
1.5	Motivations et problématiques	20
1.6	Contributions proposées	22
1.7	Plan	24
2	Les données permettant le suivi d'un étudiant	25
2.1	Données de travail	25
2.1.1	Traces d'utilisation stockées dans les LMS	25
2.1.2	Données extérieures	27
2.2	Agrégation des données en indicateurs	29
2.2.1	Indicateurs et traces d'interaction modélisées	29
2.2.2	Choix de nos indicateurs	31
3	Suivi par apprentissage automatique	43
3.1	État de l'art	43
3.1.1	Introduction aux domaines de l'apprentissage automatique et de l'exploration de données	43
3.1.2	L'exploration de données éducatives	47
3.1.3	Panorama sur les méthodes de l'exploration de données et de l'apprentissage automatique	56
3.2	Proposition d'utilisation de l'apprentissage automatique	65

4	Utilisation de HTM pour le suivi des étudiants	69
4.1	Raisons du choix de HTM-CLA	69
4.1.1	Cognitivisme et connexionnisme	69
4.1.2	Adaptation des hypothèses fondatrices de HTM à notre problème	74
4.2	L'apprentissage inspiré du cerveau humain	78
4.2.1	Fonctionnement du cerveau humain	78
4.2.2	Réseaux de neurones	82
4.2.3	Apprentissage profond	87
4.2.4	Cortical Learning Algorithm, un paradigme de compréhension et d'imitation du néocortex	95
4.3	Proposition d'utilisation de HTM-CLA	113
4.3.1	Entrées et sorties	113
4.3.2	Adaptation de HTM-CLA à nos sorties	116
4.3.3	Implémentation	120
4.3.4	Analyse des résultats obtenus	124
4.3.5	Proposition d'utilisation de HTM-CLA pour la génération de comportement	126
5	Proposition d'application dédiée au suivi des étudiants	137
5.1	Architecture	137
5.1.1	Cahier des charges	138
5.1.2	Architecture fonctionnelle	139
5.1.3	Architecture applicative	139
5.1.4	Architecture technique	141
5.1.5	Architecture orientée données	143
5.2	Conception	143
5.3	GIGA, une application dédiée au suivi des étudiants	145
5.3.1	But	145
5.3.2	Technologie	145
5.3.3	Données traitées	145
5.3.4	Fonctionnalités	146
5.3.5	Comparaison	148
5.4	Tests et résultats	150
5.4.1	Considérations de protocole	151
5.4.2	Protocole expérimental retenu	160
5.4.3	Jeux de données	160
5.4.4	Résultats	161
6	Conclusion et perspectives	167
6.1	Synthèse des contributions	167
6.2	Critiques et discussion	169
6.3	Perspectives	171
6.3.1	Modélisation et évolution de nos indicateurs	171
6.3.2	Perspectives d'évolution de GIGA	171

6.3.3	Concrétiser nos propositions utilisant HTM	173
6.3.4	Le domaine de l'ingénierie de formation	174
6.3.5	Éthique de la surveillance	175
6.4	Conclusion	176
A	Détails de l'application GIGA	177
B	Diagrammes de classes	185
B.1	HTM-CLA	185
B.1.1	Implémentation du cœur HTM	185
B.1.2	Implémentation de l'apprentissage automatique	188
B.1.3	Génération de comportement avec HTM-CLA	188
B.2	GIGA	190
B.2.1	Indicateurs	192
B.2.2	Apprentissage automatique	193
C	Fonctionnement du cerveau humain	197
C.1	Le cerveau humain	197
C.2	Le néocortex	201
C.3	Le neurone	204
C.3.1	Potentiel d'action	205
C.3.2	Types de neurones	208
C.4	Synapse	209
C.5	Plasticités et apprentissage	211
C.5.1	Plasticité synaptique	211
C.5.2	Plasticité neuronale	213
C.5.3	Plasticité cérébrale	214
C.6	Structure et assemblage des neurones dans le néocortex humain	215
C.7	La mémoire	218
C.7.1	Emplacement de la mémoire dans le cerveau	219
C.7.2	L'apprentissage	220
D	Les réseaux de neurones	221
D.1	Modèles de neurone formel	221
D.2	Méthodes d'apprentissage	222
D.3	Méthodes d'apprentissage	226
E	Test de partitionnement avec HTM-CLA	229
E.1	Le test	229
E.2	Résultats traités	229
E.3	Exemple de simulation correcte	233
E.4	Exemple de simulation incorrecte	243
F	Publications	257

Bibliographie	259
Table des figures	285

Chapitre 1

Introduction

1.1 Avant-propos

Les travaux présentés dans ce document sont essentiellement des propositions visant à résoudre le problème du suivi d'un étudiant en ligne et issues d'une vaste étude bibliographique. On y retrouvera une étude des méthodes classiquement utilisées, mais également des propositions assez innovantes issues de l'étude d'un algorithme bio-inspiré. Nos propositions sont encore assez préliminaires du fait de la grande complexité de cette partie de notre étude, qui vient de la recherche de capacités adaptatives fortes ; toutefois, elles ouvrent le champ à des travaux futurs que nous espérons nombreux dans ce domaine. Par ailleurs, nous montrons également une réalisation issue de nos travaux sous la forme d'une application web pour l'assistance au suivi des apprenants.

1.2 Contexte du projet

Cette thèse a pris la forme d'un partenariat de recherche réunissant des acteurs du monde de l'entreprise et de celui de la recherche, tous situés dans la région toulousaine, autour d'une thématique commune : les applications possibles de l'intelligence artificielle pour l'apprentissage en ligne, ou e-learning.

Ce projet a été lancé avec plusieurs objectifs. Le premier était de recenser les apports existants de l'apprentissage automatique dans le domaine de l'apprentissage en ligne et de chercher à en développer de nouveaux. Le deuxième était de créer un modèle et un moteur de prise de décision automatique pour des tuteurs virtuels basé sur la prédiction de motifs et inspiré du fonctionnement du cerveau humain, se basant sur le modèle de représentation et de prédiction de données Hierarchical Temporal Memory [Hawkins and Blakeslee, 2004]. Le troisième était de développer des applications industrielles utilisant les outils proposés pour développer les possibilités offertes par les formations en ligne.

Ce projet, que nous avons baptisé Learn Intelligence, prend la forme d'un partenariat entre l'équipe VORTEX¹ du laboratoire de recherche en informatique IRIT² et l'entreprise Andil³, spécialisée dans les technologies de l'information et de la formation. Ce partenariat est conclu autour d'une thèse CIFRE.

L'équipe de recherche VORTEX (Visual Objects from Reality To Expression) regroupe des compétences dans les domaines de l'informatique graphique, de la vision par ordinateur, de l'intelligence artificielle et des réseaux, et de façon transversale avec des travaux en réalité virtuelle et applications multimédia distribuées. Notre projet est en lien avec la composante de l'équipe qui travaille dans le domaine de la simulation comportementale et de la vie artificielle.

L'entreprise Andil, basée à Labège, dans la banlieue toulousaine, est spécialisée dans l'informatique pour l'apprentissage en ligne. Andil a notamment participé au projet national Campus de France, à l'origine de l'Université Numérique Juridique Francophone⁴ (UNJF), une des 6 universités numériques thématiques soutenue par le ministère de l'enseignement supérieur. Ainsi, Andil apporte dans ce projet une expertise associée au domaine de l'e-learning ainsi qu'aux besoins des responsables de formation en e-learning.

1.3 Le numérique et l'éducation

1.3.1 Arrivée du numérique

L'arrivée des technologies numériques et leur implantation toujours plus en profondeur dans notre société engendre évidemment de grands changements. L'introduction successive des ordinateurs, d'Internet et maintenant des smartphones, et leur démocratisation ont des impacts sur quasiment tous les aspects de notre vie quotidienne.

Un des bouleversements qui en découle est l'accès à une quantité d'information immense, mise à portée de doigts par les moteurs de recherche, et pourtant si vaste qu'elle peut devenir écrasante. Au-delà de l'accès, les technologies de l'information et de la communication (TIC) permettent également de produire et transmettre cette information.

Il semble donc naturel que les technologies numériques soient amenées à avoir une grande influence sur l'éducation, domaine où l'information joue un rôle crucial en tant que ressources d'apprentissage et, sous forme structurée, en tant que connaissances à acquérir. Cette importance est même désormais reconnue au niveau du ministère français de l'Éducation.

On définit ainsi les technologies de l'information et de la communication pour l'enseignement (TICE) comme l'ensemble des outils et produits numériques pouvant être utilisés dans le cadre de l'éducation et de l'enseignement. On n'y inclut géné-

1. <http://irit.fr/-Site-de-l-equipe-Vortex->

2. <http://irit.fr/>

3. <http://www.andil.fr/>

4. <http://www.unjf.fr/>

ralement pas les outils périphériques comme les logiciels de gestion administrative, même si les TIC ont également fourni de nombreux outils de ce type.

Le domaine de recherche associé à ces évolutions et ces usages s'appelle EIAH. Ce sigle signifie « Environnement informatique pour l'apprentissage humain ». [Tchounikine, 2002] en propose la définition suivante : « Un EIAH au sens large est un environnement qui intègre des agents humains (e.g. élève ou enseignant) et artificiels (i.e., informatiques) et leur offre des conditions d'interactions, localement ou à travers les réseaux informatiques, ainsi que des conditions d'accès à des ressources formatives (humaines et/ou médiatisées) locales ou distribuées ».

L'EIAH est un domaine de recherche pluridisciplinaire. On y retrouve les disciplines de la pédagogie, la didactique, la psychologie cognitive, les sciences de l'éducation et l'informatique. Ainsi, la conception et l'ingénierie des EIAH doivent être abordés comme des problèmes transdisciplinaires. En outre, science et technique sont inextricablement liées dans le domaine des EIAH : le système étudié est à la fois la source et le but du travail scientifique, et l'expérimentation joue un rôle crucial.

Dans un EIAH, l'outil informatique peut avoir divers rôles non mutuellement exclusifs : celui de présentation de l'information (par exemple un hypermédia ou un LMS), d'outil de traitement de connaissances (par exemple un tuteur virtuel intelligent) ou d'outil de communication.

Le numérique ne doit pas être perçu comme une nouvelle discipline d'enseignement, mais bien plutôt comme une dimension transversale et sous-jacente à tous les enseignements. Son usage est perçu comme nécessaire dans la mesure où il sous-tend complètement la société actuelle : « Avant, nous allions sur Internet, maintenant, nous sommes dedans. » [Rognetta et al., 2012]

Au Québec, une étude du CEFRIO fait ce constat : « Les TIC sont omniprésentes dans la vie des élèves et des étudiants du Québec : ceux-ci s'en servent continuellement pour se divertir, pour communiquer avec leurs amis ou pour faire leurs devoirs. En fait, les TIC sont partout... sauf dans les salles de classe! [...] On ne peut plus leur enseigner comme on le faisait avant qu'ils découvrent la puissance du numérique comme outil de recherche d'informations et de collaboration. » [Tanguay et al., 2011]

1.3.2 Outils

L'utilisation du numérique pour l'éducation se fait via divers outils, produits et services. Nous en proposons ici une liste, sans doute non exhaustive, suivant notre propre catégorisation.

Une première classe d'outils concerne le matériel utilisable. Les enseignants peuvent utiliser des tableaux blancs interactifs peuvent enrichir une présentation en couplant un vidéoprojecteur à une commande interactive tactile utilisant les doigts ou un stylet, mais aussi des outils d'enregistrement vidéo (une caméra classique ou les lunettes Google Glass) pour s'auto-analyser dans leurs interactions avec les élèves. Les élèves peuvent utiliser des ordinateurs, connectés ou non à Internet, mais aussi des tablettes tactiles, particulièrement bien adaptées aux élèves de maternelle et primaire, du fait de leur interface intuitive ne requérant que peu de lecture. Une

formule nommée *Bring Your Own Device (BYOD)* (apportez votre propre appareil) suggère l'utilisation par les élèves de leurs propres appareils (smartphones, tablettes, ordinateurs portables. . .) pour les activités scolaires. Malgré le gain de coût pour les institutions, cette formule peut poser problème aux plus pauvres qui ne seront peut-être pas assez bien équipés.

Une deuxième classe d'outils concerne l'utilisation pour l'éducation de logiciels et services génériques. Ainsi, les réseaux sociaux peuvent relier entre eux les différentes personnes intéressées par un même sujet. Diverses plate-formes contiennent des contenus éducatifs de haute qualité : Wikipedia, TED, Youtube. . . Des simulateurs ou systèmes experts peuvent être utilisés à des fins pédagogiques pour mieux comprendre un sujet donné.

Les ressources numérisées représentent une troisième catégorie. Les hypermédias sont des médias riches offrant une navigation entre diverses ressources multimédias. Ils sont utilisés comme ressources riches d'apprentissage. Les présentations assistées par ordinateurs peuvent aider les enseignants à transmettre une notion. Les manuels scolaires peuvent devenir entièrement numériques, ce qui diminue le poids du cartable et le coût du matériel, et permet une navigation facilitée, avec par exemple des outils de recherche.

Une quatrième classe d'outils est représentée par les plate-formes permettant l'accès des utilisateurs à certaines ressources. Les plate-formes d'apprentissage en ligne, ou LMS, comme Moodle, Sakai ou Claroline, fournissent contenus et activités pour l'apprentissage en ligne. Elles proposent parfois des dispositifs d'apprentissage collaboratif. Les formations en ligne ouvertes à tous, ou MOOC, se distinguent des précédentes par le grand nombre d'utilisateurs visés. Un ENT (environnement numérique de travail) est un un portail qui centralise des informations en lien avec une formation et permet un accès à distance de ressources numériques. Ce terme est couramment utilisé dans les collèges, les lycées et les universités françaises pour désigner un portail intranet, à l'usage des élèves et du personnel, ou un portail extranet à destination également des parents. On parle également de cartable en ligne.

Enfin, les logiciels spécialisés pour l'éducation constituent une cinquième classe d'outils. Les jeux sérieux d'apprentissage combinent une intention pédagogique avec des mécanismes ludiques (interactivité, objectifs à atteindre. . .) pour rendre l'apprentissage plus attrayant. Les exercices peuvent automatiser la réalisation et la correction d'exercices, voire même leur génération à partir d'une base fournie. Les micro-mondes sont des environnements informatiques permettant un apprentissage actif dans un domaine spécialisé. Les plus connus sont le langage de programmation Logo et le logiciel de géométrie Cabri. Les tuteurs intelligents sont utilisés pour faire des retours immédiats et personnalisés aux apprenants quand ils réalisent des activités.

Comme pour toute autre technologie numérique se pose le problème de l'utilisation de logiciels et ressources libres, qui sont à préférer quand elles sont disponibles. Ils permettent en effet à chaque utilisateur de personnaliser l'outil suivant son besoin et de mettre ses contributions à disposition de tous. On peut notamment citer les outils

suivants : les LMS Moodle, Claroline et Dokeos, l'encyclopédie en ligne Wikipedia, la galaxie de services et ressources Framasoft, les cartes électroniques Arduino, le logiciel de gestion de tableau blanc interactif Sankoré, le navigateur Firefox, les logiciels génériques OpenOffice/Libre Office, Audacity, Blender, Gimp, Inkscape, le système d'exploitation Edubuntu qui comprend nativement les suites éducatives KDE Education et Gcompris, le serveur pour réseau scolaire AbulÉdu, et bien d'autres encore.

1.3.3 Enjeux

On retrouve bon nombre d'enjeux très importants dans la problématique de l'entrée du numérique dans l'éducation.

Un premier enjeu est lié à l'adaptation de la pédagogie au numérique. On pourrait se contenter de l'utilisation des outils numériques en remplacement d'anciennes méthodes, mais on peut également les intégrer de façon suffisamment fondamentale pour pouvoir se remettre en question et évoluer grâce à eux. En effet, il commence à être montré que la simple utilisation d'outils numériques ne donne aux élèves qu'un regain de motivation temporaire, probablement dû à un attrait pour la nouveauté. En revanche, si l'utilisation de ces outils peut se traduire par un apprentissage plus interactif ou plus personnalisé, on peut alors assister à un engagement plus durable des élèves. Elle peut également engager davantage les parents en les associant au projet éducatif. Un deuxième aspect de cet enjeu concerne donc les résultats de cette pédagogie sur les élèves. Il conviendra de chercher à comprendre et mesurer ses impacts sur l'échec scolaire, l'insertion dans la vie professionnelle et la formation à l'utilisation du numérique.

Un deuxième enjeu concerne l'équipement matériel nécessaire (ordinateurs, tablettes, tableaux blancs interactifs...). Cet équipement a nécessairement un coût, qui doit pouvoir être soutenu par la communauté, et risque de créer des inégalités entre zones géographiques riches et pauvres, entre zones urbaines très connectées et zones rurales manquant d'infrastructures. Il peut ainsi maintenir les inégalités sociales. À l'inverse, si toutes les écoles parviennent à s'équiper, une formation en présentiel pourrait devenir un luxe du fait du coût du salaire des intervenants par rapport à celui d'un investissement en matériel et ressources déjà rentabilisé. Par ailleurs, l'acquisition de cet équipement peut avoir pour aspect positif la réduction du poids des cartables des élèves, en numérisant les manuels, les cahiers de leçons et d'exercice.

Un troisième enjeu concerne les enseignants, qui risquent de voir leur rôle modifié par la modification de la pédagogie. Il faudra alors voir si cette remise en cause est perçue comme valorisante ou non, et s'il ne faut pas alors leur proposer un accompagnement ou modifier leur formation.

1.3.4 La pédagogie à l'épreuve du numérique

Nous allons ici développer notre premier enjeu, qui nous semble particulièrement important puisque, comme nous l'avons mentionné, la remise en cause de la pédagogie initiée par l'arrivée du numérique peut avoir de grands effets sur l'engagement des élèves dans leur formation, et donc dans leur succès ultérieur.

La simple utilisation d'outils numériques n'est pas suffisante à un réel succès, comme le montrent les citations suivantes :

« Il ressort des recherches, études, analyses et discours de toutes sortes, qu'il ne faut pas chercher dans les technologies la recette de l'élévation du taux de réussite des apprenants. Les facteurs de réussite sont ailleurs : d'une part, dans la personnalité de l'apprenant et, d'autre part, dans l'art du pédagogue qui le guide et l'accompagne. » [Loisier, 2011]

« Le facteur à considérer n'est pas tant la technologie comme telle que l'intention pédagogique qu'elle permet de réaliser [...]. Il ne s'agit donc pas réellement d'évaluer l'impact de l'utilisation des TIC, mais plutôt l'impact d'une stratégie pédagogique qui a recours à telle ou telle technologie adaptée à ses intentions. » [Barrette, 2004]

Il faut donc éviter que, d'enthousiastes et motivés que les enseignants peuvent débiter, ils ne deviennent technocentristes, c'est-à-dire « des personnes qui peuvent difficilement s'empêcher de conclure que les TIC sont le "remède" à la majorité des maux que connaît l'école de nos jours » [Viau, 2005]. En effet, le consensus parmi les chercheurs en pédagogie s'oriente plutôt pour dire que ce sont les stratégies pédagogiques permises par les TICE qui font une réelle différence tant au niveau de la réussite que de la motivation des élèves. Et elles sont nombreuses : on peut dire que la communauté de réflexion pédagogique est en plein essor depuis quelques années. Nous allons essayer de lister quelque-unes des idées phares de ce renouveau.

L'apprentissage actif cherche à donner aux étudiants un rôle plus actif, comme son nom l'indique, dans le processus d'apprentissage. Il est à mettre en contraste avec le cours magistral, où son rôle se borne à une écoute passive d'un expert. Pour le mettre en œuvre, on demande souvent aux élèves de se lancer dans des discussions collectives et des activités individuelles ou de groupe. Ils ont également une plus grande autonomie dans l'organisation de leur travail.

Des études ont montré que cette forme d'apprentissage était plus efficace : dans [Freeman et al., 2014], 34% des étudiants ayant suivi un cours magistral ont échoué leur examen, pour 22% des étudiants d'un cours avec apprentissage actif, comme on peut le constater dans la figure 1.1⁵. La moyenne des notes obtenue était également meilleure d'environ 6%.

Pour rendre l'apprentissage plus actif, on peut mettre en œuvre une stratégie de pédagogie inversée ou classe inversée, qui consiste à utiliser le temps de classe pour les travaux pratiques en repoussant le cours magistral comme devoir à la maison. Dans de nombreux cas, le cours magistral est enregistré sous la forme d'une vidéo que les élèves ont pour consigne de regarder. Un autre intérêt de cette méthode est de permettre à l'élève d'apprendre à son rythme, puisqu'il peut contrôler la vidéo.

5. Source : [Freeman et al., 2014]

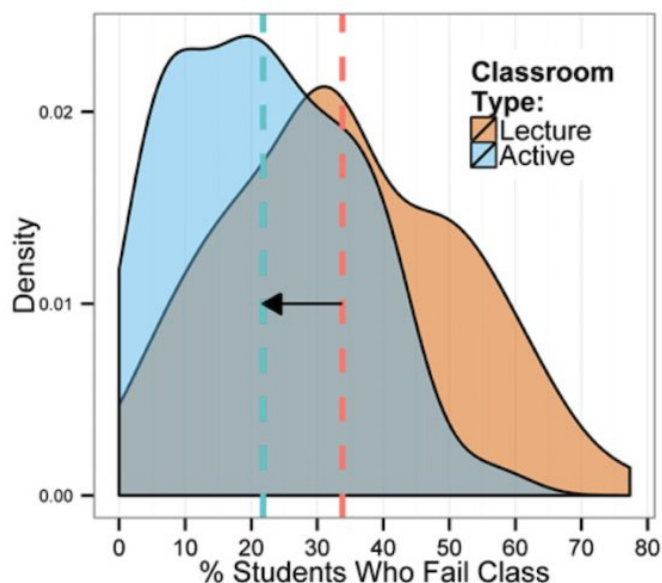


FIGURE 1.1 – Le taux de succès des élèves en apprentissage actif ou passif

L'aspect collaboratif de l'apprentissage est également très important, puisque des élèves travaillant en groupes obtiennent de meilleurs scores que lors d'un travail individuel. Toutefois, si le groupe est trop homogène, c'est le résultat inverse qui se produit. [Johnson and Johnson, 1980].

Pour les enseignants également, la collaboration produit de nombreux effets positifs. De nombreux enseignants intéressés par les nouvelles formes de pédagogie se regroupent sur les réseaux sociaux, par exemple Twitter, pour partager leurs expériences et échanger leurs idées. Au-delà de cette ressource, qui leur apporte également du recul sur leur pratique, cela peut leur donner l'impression précieuse d'appartenir à une communauté de collègues de même sensibilité.

L'apprentissage peut également de plus en plus devenir informel et se faire en dehors de l'école. Le savoir est présenté dans des vitrines telles que Wikipedia et les conférences TED, permettant à chacun de se renseigner sur un sujet qui l'intéresse. Ces sources diverses participent aussi à la remise en cause des rythmes scolaires actuels, puisque l'e-learning permet un apprentissage à tout moment et tout endroit, ainsi que plus modulaire. Par ailleurs, la théorie de l'apprentissage social postule que l'enfant apprend davantage par imitation qu'avec des consignes détaillées.

Le rôle de l'enseignant est donc nécessairement modifié. Il n'est plus la seule source du savoir, surtout avec l'émergence des MOOCs, et son rôle n'est plus nécessairement sa diffusion. Il doit donc devenir davantage un facilitateur d'apprentissage, choisissant les objectifs pédagogiques et les ressources et les mettant en scène, sachant répondre aux questions des élèves et leur transmettant son enthousiasme.

En parallèle se construisent des réflexions qui ont moins à voir avec le numérique, comme une remise en question des notes, qui ont plusieurs défauts :

- elles transforment la motivation intrinsèque des élèves en motivation extrinsèque
- elles ne reflètent pas le vrai niveau des élèves, comme l'a montré André Antibi avec sa notion de constante macabre [Antibi and Luciani, 2003]
- elles ne traduisent pas bien les compétences réellement acquises ou non par les élèves.

Toutefois, là encore, le numérique offre des opportunités d'expérimentation, par exemple en facilitant l'évaluation anonyme des élèves entre eux.

On peut donc dire que le numérique offre une opportunité unique pour refonder l'école et la pédagogie. En offrant de nouvelles possibilités, il a libéré une réflexion très riche et innovante, et démocratisé des pratiques jusque-là restées expérimentales.

1.4 L'apprentissage en ligne

L'apprentissage en ligne (*e-learning* en anglais) vise plus spécifiquement à utiliser les technologies de l'Internet pour permettre l'accès à des ressources et services éducatifs, le plus souvent multimédias, et également pour favoriser les échanges et la collaboration avec les tuteurs et les pairs, qui sont souvent physiquement distants.

Historiquement, il a d'abord concerné la formation continue, l'enseignement supérieur et la formation en entreprise. Il est en effet mieux adapté à un public adulte puisqu'il nécessite autonomie et responsabilité de la part de l'apprenant.

Comme nous allons particulièrement nous intéresser dans le cadre de notre projet à l'apprentissage en ligne, nous avons jugé utile de préciser ici quelques-unes de ses spécificités.

1.4.1 Terminologie

L'apprentissage mixte (*blended learning* en anglais) combine apprentissage en ligne, ou distanciel, et apprentissage classique hors ligne, qui est alors qualifié de présentiel. Ces présentiels peuvent servir à interagir avec le tuteur et les pairs, et parfois à présenter ses travaux et son progrès.

Parmi les outils de communication en ligne, on peut distinguer les outils synchrones, où la communication se fait en temps réel (chat, visioconférence) des outils asynchrones (forum, e-mail).

Une plate-forme d'apprentissage en ligne (en anglais *learning management system* (LMS)) est un logiciel web permettant l'apprentissage en ligne. Il regroupe donc des contenus, activités et ressources éducatives, des fonctionnalités de télé-tutorat et de communication, et des outils administratifs. Les LMS les plus utilisés sont Moodle et Sakai. Un LCMS, ou *Learning Content Management System*, a en outre la capacité d'édition de contenu.

L'ingénierie pédagogique est l'activité d'étudier, de concevoir, de réaliser et d'adapter des dispositifs d'enseignement.

Une formation en ligne ouverte à tous (FLOT), aussi appelée cours en ligne ouvert et massif (CLOM), (en anglais : *massive open online course*, MOOC), est une formation uniquement en ligne, qui a pour spécificité son grand nombre de

participants : dans le monde anglophone, il peut arriver que plus de 100 000 personnes soient réunies pour un cours. Il en existe deux grands types : les xMOOC (le x signifie *extended*), qui sont généralement des transpositions en ligne de cours existant hors ligne et délivrent fréquemment un certificat de réussite à la fin, et les cMOOC (le c signifie *connectivist*), qui mettent l'accent sur l'apprentissage collaboratif entre pairs, avec par exemple des corrections par les pairs et du travail de groupe.

Les MOOC ont connu un essor important en 2012, avec le lancement de deux cours à grand succès par l'université de Stanford. Les plate-formes MOOC les plus connues sont américaines. On recense Udacity (1,6 million d'utilisateurs), Coursera (9,2 millions d'utilisateurs) et edX (2,4 millions d'utilisateurs). En France, une plate-forme MOOC basée sur edX et nommée FUN (France Université Numérique) a ouvert en octobre 2013 (200 000 utilisateurs).

1.4.2 Avantages et inconvénients

Les avantages souvent cités sont :

- un moindre coût : pas systématiquement, mais cela peut être le cas en raison du moins grand besoin de personnel et de la réutilisation de ressources. Notamment, un grand nombre de MOOC sont gratuits.
- une gestion du temps personnalisable et flexible, ce qui est notamment un avantage pour la formation continue ou les étudiants devant travailler pour financer leurs études.
- une possibilité de travailler depuis n'importe quel endroit, ce qui peut être pratique notamment en situation de handicap, et permet de réduire l'impact du transport en termes de coût et de bilan écologique.
- la possibilité d'atteindre davantage d'étudiants, notamment avec les MOOC.
- la disponibilité d'un large panel d'approches pédagogiques, dont certaines très innovantes. un apprentissage de la responsabilisation des étudiants.
- l'acquisition d'un savoir-faire lié aux technologies employées.
- une possibilité d'évaluation automatique de certaines activités, libérant du temps pour les tuteurs.
- un accès aux meilleurs experts du domaine.
- une meilleure interaction avec les tuteurs et les pairs via les outils de communication offerts.
- l'apprentissage mixte permet de mettre en œuvre une classe inversée, dont les effets sont bénéfiques grâce à l'apprentissage actif.

L'apprentissage en ligne peut toutefois présenter aussi certaines limites :

- l'apprenant peut être plus facilement démotivé, procrastiner ou manquer d'assiduité par manque d'interaction directe et de sociabilisation (un bon tutorat permet d'éviter ce risque). Notamment, les MOOC entièrement en ligne ont des taux d'abandon supérieurs à 90%.
- il nécessite un bon accès à Internet.
- il nécessite des compétences technologiques.
- de nombreuses formations et ressources ne sont disponibles qu'en anglais, alors

- qu'il est plus facile d'apprendre dans sa langue maternelle.
- il peut être tentant de ne proposer que peu ou pas de tutorat. C'est souvent le cas des MOOCs, qui ont un très grand nombre d'élèves, mais pas forcément davantage d'enseignants.
 - il peut être plus facile de tricher hors de la vue du tuteur.
 - l'apprentissage est à la merci de difficultés techniques.
 - il peut être tentant de remettre à plus tard une formation qu'on sait disponible en permanence sur Internet, mais cela peut jouer des tours en situation de crise ou d'urgence.
 - une formation uniquement en ligne n'est évidemment pas adaptée à tous les savoir-faire, dont un bon nombre nécessite de la pratique.
 - il est peu écologiquement soutenable, nécessitant une consommation électrique, des appareils utilisant des métaux rares qui sont rapidement obsolètes et encore trop mal recyclés.

1.5 Motivations et problématiques

Nous avons vu qu'un des grands problèmes de l'apprentissage en ligne est la démotivation des étudiants. Si l'étudiant n'est pas régulièrement suivi et remotivé par des interactions avec ses tuteurs ou ses pairs, il risque d'abandonner la formation, ou de ne pas présenter une assiduité suffisante pour la réussir. Par ailleurs, s'il ne se rend pas bien compte de la charge de travail à accomplir, il peut la remettre au dernier moment, ce qui augmente également ses risques d'échec. Plus la proportion de la formation se déroulant en ligne est grande, plus ces risques augmentent, comme le montrent les statistiques d'abandon des élèves inscrits à un MOOC.

Il a été montré que des étudiants démotivés peuvent se livrer à des comportements nuisant à leur apprentissage, comme par exemple une utilisation abusive d'un système [Baker et al., 2004b], ou un parcours trop rapide ou très lent et inefficace des ressources pédagogiques [Cocea and Weibelzahl, 2011, Carini et al., 2006]. En retour, ces étudiants apprennent moins que les autres. Par ailleurs, l'auto-évaluation de la motivation des étudiants est un bon indicateur de leurs stratégies d'apprentissage [Beal et al., 2006].

Le soutien à la motivation des étudiants peut prendre des formes très simples, comme l'apport d'une interface graphique plus jolie ou plus interactive, ou une forme plus ludique, qui ont été testés avec succès par [Barolli et al., 2006, Chen et al., 1998]. Une deuxième approche, plus compliquée à mettre en œuvre, se base sur la personnalisation des contenus ou des approches [Tseng et al., 2008]. Toutefois, elle réclame non seulement des évolutions logicielles, mais également un grand travail de modélisation des élèves, parfois jugé trop important.

L'approche que nous privilégions est une troisième voie, qui consiste à chercher à détecter le manque de motivation, et privilégier la réaction par contact humain. Nous pensons que si certaines situations d'échec sont peut-être inéluctables (circonstances personnelles, mauvaise adaptation des désirs de l'apprenant à la formation proposée),

une bonne partie d'entre elles pourraient être évitées par un repérage précoce qui donnerait lieu à un recadrage et un suivi plus attentif et personnalisé de la part des tuteurs ou des responsables de formation. Il nous paraît important que ce repérage arrive le plus tôt possible, dès qu'un élève commence à décrocher, pour qu'il ne prenne pas trop de retard par rapport aux autres dans le délai de l'intervention des tuteurs. L'action de soutien du tuteur peut alors arriver au bon moment. Il est important également qu'elle soit personnalisée, pour s'adapter à la situation particulière de l'étudiant.

Notre objectif principal avec ce projet est donc d'améliorer les possibilités de suivi des élèves en cours de formation en ligne pour éviter leur décrochage ou leur échec.

Or, la mesure de la motivation des étudiants en ligne est un vrai défi. En effet, dans un cours en présentiel, l'instructeur peut se baser sur le langage du corps de ses étudiants. S'il les voit regarder dans le vague, discuter entre eux ou gigoter, il peut intervenir, par exemple en s'adressant directement aux élèves ou en changeant de sujet. Rien de tel n'est disponible pour les cours en ligne, dont les enseignants doivent donc trouver d'autres ressources.

Pour ce faire, nous pensons nous baser notamment sur l'étude des traces d'activité modélisées sous forme d'indicateurs, comme dans [Lafraquière and Prié, 2007] et [Choquet et al., 2007]. Or, à ce jour, le LMS que nous utilisons le plus couramment, Moodle [Moodle Trust, 2013, Cole and Foster, 2007], ne dispose que de quelques statistiques difficiles à examiner et analyser. Notamment, ces statistiques sont uniquement individuelles, ce qui empêche d'avoir une vision globale d'un groupe d'élèves et de pouvoir les comparer entre eux. En outre, il ne permet que d'avoir accès à une suite de logs d'activité sans guère proposer de vision plus synthétique, hormis quelques graphiques pour les données de connexion.

Via Andil, nous avons pu connaître les pratiques de personnes chargées du suivi des élèves. Nous avons constaté que, pour les raisons que nous venons d'évoquer, elles s'en tenaient à deux seuls indicateurs du succès d'un étudiant : la date de sa dernière connexion et le nombre d'activités notées qu'il avait réalisées. Une pratique courante était d'accéder à ces statistiques individuelles, puis de les reporter manuellement dans un tableur afin de pouvoir garder une vision d'ensemble du groupe et de ne pas avoir à répéter trop souvent cette manœuvre longue et pénible. Cela nous a semblé dommage en regard des vastes possibilités offertes par l'enregistrement que Moodle fait de chaque clic d'un apprenant.

Nous avons aussi mentionné que ce projet a également été lancé avec l'objectif d'étudier les possibilités de l'algorithme Hierarchical Temporal Memory (HTM) [Hawkins and Blakeslee, 2004], dans sa version Cortical Learning Algorithm (CLA) [Numenta, 2010], que nous abrègerons par la suite en HTM-CLA. Cet algorithme connexionniste innovant s'inspire du fonctionnement du cerveau humain, à un niveau de réalisme plus important que celui des réseaux de neurones classiques, et en réutilisant des principes de l'apprentissage profond, pour proposer des fonctionnalités de représentation et de prédiction de données par un apprentissage incrémental non supervisé.

Cet algorithme s'inscrit en lien avec les sujets étudiés par l'équipe Vortex, qui cherche à utiliser des méthodes bio-inspirées, à des fins de simulation bio-réaliste ou de proposition d'algorithmes innovants. Ainsi, [Panzoli, 2008] a introduit dans l'équipe l'étude de réseaux de neurones pour la génération de comportement. Par ailleurs, une de nos contributions lors de ce travail a été de montrer la bonne adaptation des hypothèses sous-jacentes de cet algorithme à notre problématique, justifiant ainsi a posteriori notre intérêt, comme nous le verrons par la suite.

Notre deuxième objectif dans le courant de cette thèse était donc d'analyser l'adaptation de cet algorithme à notre problématique du suivi d'étudiants. Nous envisagions trois façons de l'utiliser : une première en l'adaptant à des fonctionnalités classiques de type partitionnement, classification, régression et prédiction ; une deuxième en l'utilisant comme base d'un moteur de génération de comportement, qui pourrait être utilisé pour créer un tuteur virtuel intelligent chargé de conseiller les apprenants en temps réel ; une troisième en lui donnant à apprendre les contenus pédagogiques eux-mêmes pour étudier la représentation qu'il en aurait construite. Nous n'avons toutefois eu le temps d'examiner que les deux premières propositions.

1.6 Contributions proposées

Nous avons étudié les données disponibles pour le suivi des étudiants et établi une réflexion sur le choix d'attributs d'apprentissage qui nous semblent à la fois pertinents à notre projet et suffisamment génériques pour pouvoir être utilisés par d'autres.

Nous avons cherché à comprendre comment l'apprentissage automatique et l'exploration de données étaient utilisés dans le domaine de l'apprentissage en ligne, et comment ils pouvaient plus spécifiquement être adaptés à la problématique du suivi des étudiants.

Nous avons étudié les spécificités de l'algorithme HTM-CLA et en avons conclu qu'il paraît particulièrement bien adapté à notre problématique. En conséquence, nous avons conçu des solutions pour adapter HTM-CLA à des fonctionnalités classiques d'apprentissage automatique pour pouvoir tester ses performances. Dans une autre proposition, nous suggérons de l'adapter à la génération de comportement, ce qui pourrait servir de base à un tuteur virtuel intelligent. Pour cela, nous avons étudié quelques modèles similaires de génération de comportement intelligent.

Nous avons proposé une architecture d'application permettant de regrouper et traiter ces données, calculer ces indicateurs et traiter ces données par apprentissage automatique comme nous l'avons proposé. Nous avons ensuite, en partenariat avec Andil, réalisé cette application.

Nous avons, grâce à cette application, pu mener de premières expériences de partitionnement de données sur des données issues de formations réelles, grâce à un institut de formation partenaire d'Andil, JurisCampus⁶. Nous présentons les résultats de ces expériences accompagnés d'une analyse.

6. <http://juriscampus.fr/>

Nos propositions sont résumées dans le schéma 1.2.

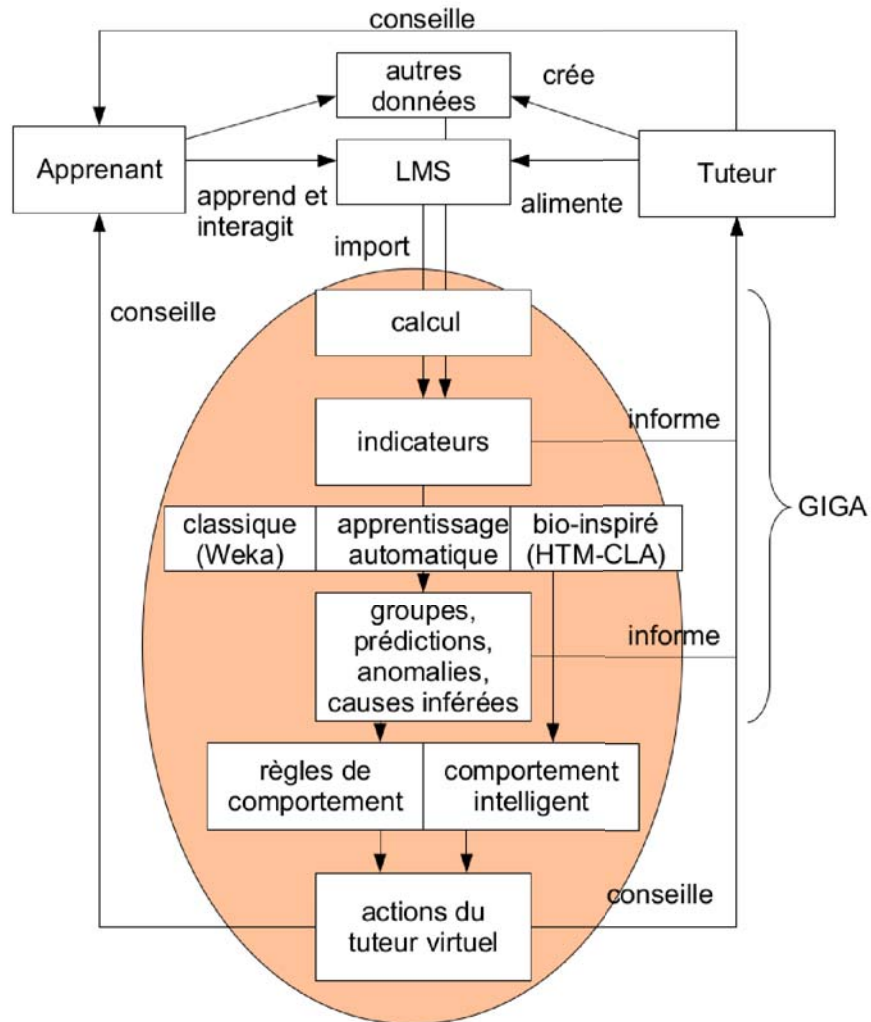


FIGURE 1.2 – Notre proposition d'architecture de données pour GIGA

1.7 Plan

La suite du mémoire prendra la forme suivante.

Le chapitre 2 sera dédié aux données dont nous disposons pour analyser le comportement des apprenants. Nous y présenterons également la façon dont ces données peuvent être agrégées pour réaliser des indicateurs adaptés à la fois aux données disponibles et à notre problématique de suivi.

Le chapitre 3 commence par rappeler les principes de l'apprentissage automatique et de l'exploration de données. Elle présente ensuite un rapide bilan de l'utilisation de ces méthodes dans le domaine de l'apprentissage en ligne, qui est suivi par nos propositions de son utilisation dans le domaine plus précis du suivi des étudiants.

Dans le chapitre 4, nous présentons l'algorithme HTM-CLA, auquel nous nous sommes particulièrement intéressés, après un rappel des connaissances sur le cerveau humain et des divers algorithmes d'apprentissage automatiques qui s'en sont également inspirés. Nous expliquons ensuite pourquoi nous pensons que cet algorithme pourrait être particulièrement bien adapté à notre problématique, et proposons des adaptations nécessaires pour pouvoir l'utiliser à ces fins, dont nous détailleront également les possibles implémentations.

Le chapitre 5 présente GIGA, l'application que nous avons réalisée en partenariat avec Andil pour mettre en œuvre nos propositions, tout en détaillant ses choix architecturaux, et se termine par la présentation de résultats de tests de partitionnement de données que nous avons pu mener via cette application.

Pour conclure, le chapitre 6 revient sur les contributions de cette thèse et évoque des pistes pour poursuivre notre projet.

En annexe A, nous présentons davantage de captures d'écran de GIGA afin de détailler ses fonctionnalités. L'annexe B apporte des détails d'implémentation de nos propositions, avec notamment des diagrammes de classes. Une étude plus détaillée du fonctionnement du cerveau humain peut permettre au lecteur intéressé de mieux comprendre les fondements biologiques de l'algorithme HTM-CLA ; il la trouvera alors en annexe C. De même, l'annexe D offre un complément d'état de l'art dans le domaine des réseaux de neurones. L'annexe E vise à montrer un exemple de simulation de partitionnement obtenu avec notre implémentation de HTM-CLA. Enfin, l'annexe F récapitule nos publications.

Chapitre 2

Les données permettant le suivi d'un étudiant

Dans ce chapitre, nous allons commencer par analyser les données dont nous disposons pour mieux comprendre le comportement des apprenants. Nous verrons que si la source principale de ces données reste le LMS, des données issues d'autres sources peuvent également être très intéressantes. Ces données peuvent être agrégées pour réaliser des indicateurs du comportement des étudiants, qui nous seront utiles dans le cadre de notre problématique de suivi. Après avoir défini via l'état de l'art du domaine le processus de construction de tels indicateurs, nous proposerons notre liste, que nous comparerons à celles issues de travaux similaires.

2.1 Données de travail

Nous allons ici chercher à recenser les données à notre disposition pouvant être utiles pour le suivi des apprenants.

2.1.1 Traces d'utilisation stockées dans les LMS

Les LMS sont une des plus importantes sources de données quant au comportement d'un élève en formation e-learning. En effet, les principaux LMS stockent des traces numériques d'un grand nombre des actions effectuées par les utilisateurs de la plate-forme dans leur base de données. Majoritairement, ces traces prennent la forme d'un historique des actions. Nous allons détailler les capacités de ce stockage de traces en étudiant de plus près l'exemple de Moodle, sur lequel nous avons travaillé. Notons toutefois que les autres LMS offrent généralement des traces similaires en nature, même si le détail de leur stockage peut différer.

Les historiques enregistrés par la plate-forme Moodle à chaque action de ses utilisateurs sont stockés dans une seule table de leur base de données intitulée log, qui contient :

- la date et l'heure de l'action
- le numéro d'identifiant de l'utilisateur qui a effectué cette action
- l'adresse IP de cet utilisateur à ce moment donné
- l'identifiant correspondant à l'action effectuée
- d'éventuelles informations supplémentaires, qui sont différentes suivant le type de l'action.

Time	IP Address	Full name	Action	Information
Mon 1 February 2010, 06:12 PM	65.41.7.67		resource view	Click here for the "MNT Certificate"
Mon 1 February 2010, 06:12 PM	65.41.7.67		course view	MNT Module 7. Intro to Wound Healing
Mon 1 February 2010, 06:12 PM	65.41.7.67		quiz view	MNT Final Assessment
Mon 1 February 2010, 06:11 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:11 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:10 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:10 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:08 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:07 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:07 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:07 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz review	MNT Final Assessment
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz close attemp	MNT Final Assessment
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:05 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:04 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:01 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 06:00 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:59 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:57 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:57 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:55 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:54 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:52 PM	65.41.7.67		quiz continue attemp	13
Mon 1 February 2010, 05:51 PM	65.41.7.67		quiz view	MNT Final Assessment

FIGURE 2.1 – Les traces enregistrées par Moodle

Grâce à ce système de traces, nous pouvons savoir quelles ressources un étudiant a consultées et quand, et en connaître le résultat quand il s'agit d'un exercice évalué.

Les données brutes fournies par le système de traces de Moodle qui nous semblent particulièrement intéressantes à étudier sont donc les suivantes :

- le début et la fin de session d'un étudiant
- la lecture d'une leçon, en ligne ou sous forme de PDF
- la consultation d'une ressource associée à une leçon (bibliographie, glossaire, etc.)
- la réalisation d'activités non notées (sondage, journal, blog, wiki...)
- la réalisation d'activités notées (quiz, SCORM, devoir, Hot Potatoes, atelier...)
- les notes qui ont été obtenues à ces activités notées
- les messages lus et écrits sur le forum et le chat
- la participation à toute autre forme de communication entre étudiants, enseignants et personnel administratif (greffon de visioconférence...)

À toutes ces données serait associée leur horodate.

En effet, le moment de début et de fin de session peuvent nous indiquer à quel moment l'étudiant était en ligne. On peut s'en servir pour calculer le temps qu'il passe à ses activités d'apprentissage, mais également à quel moment de la journée, de la semaine, et du calendrier de la formation il réalise ces activités. Il pourrait être intéressant d'en repérer des motifs : ne travaille-t-il qu'en soirée ? Que juste avant les deadlines ?

Il est bien évidemment important de savoir si un étudiant a étudié ou non les contenus pédagogiques mis à sa disposition, que ce soient les leçons ou les ressources associées. À partir de ces logs, nous pouvons savoir quelles leçons et ressources ont été lues, mais également lesquelles ne l'ont pas été ; nous pouvons également connaître la temporalité de cette activité ; et nous pouvons corrélérer ces données à d'autres pour savoir si, par exemple, l'étudiant a bien entièrement lu la leçon et les ressources associées avant d'aller réaliser le QCM associé, ou pas. C'est le même genre de connaissance que nous pouvons déduire des logs concernant la réalisation d'activités notées ou non notées (réalisation ou pas, temporalité, corrélation avec les autres données).

Au début du projet, nous avons hésité à inclure dans sa portée les données de type évaluatif, c'est-à-dire les notes et résultats obtenus à diverses activités. Nous nous sommes en effet dit qu'il pouvait ne s'agir que de suivre l'activité et l'implication d'un étudiant, et que la connaissance de ses résultats intermédiaires pouvait biaiser ce suivi. Après réflexion, nous avons décidé de les inclure tout de même afin d'obtenir un autre type d'information. En effet, cela peut nous permettre de savoir si un étudiant déploie son activité de façon efficace ou non, tel que retranscrit dans ses évaluations. Un étudiant très actif mais n'obtenant que de mauvaises notes nous semble devoir susciter une interpellation et une éventuelle intervention de ses encadrants au même titre qu'un manque d'activité.

Enfin, les données de participation sociales peuvent nous permettre de savoir à quel point l'étudiant se base sur les interactions pour son apprentissage, de savoir à quel moment il le fait et de corrélérer cela avec les autres données pour savoir si l'aspect social de son apprentissage a un impact sur d'autres aspects.

2.1.2 Données extérieures

Toutefois, se limiter aux traces stockées par le LMS serait dommageable, dans la mesure où bon nombre de données pertinentes au suivi des étudiants n'y figurent pas. Certaines en sont absentes par limitation technique : ainsi, par exemple, sur Moodle, on ne peut pas connaître avec précision les durées des sessions, car rien n'informe si un étudiant a quitté la page sans passer par le bouton de déconnexion de la plate-forme. On peut donc calculer des durées de session à partir de la durée d'expiration de la session, mais cette estimation sera forcément très imprécise, d'autant plus que la durée d'expiration est importante.

On peut aussi imaginer de rajouter au système des données qui ne figurent pas encore dans le LMS, mais pourraient y figurer, dans la mesure où elles sont caractéristiques de la formation qui y est proposée. Ainsi, il est possible d'ouvrir certains cours au fur et à mesure de la formation. Actuellement, Moodle ne stocke pas la date d'ouverture du cours, ce qui est très ennuyeux dans la mesure où cela ne permet pas de faire la différence entre un étudiant qui s'est intéressé à un cours dès son ouverture tardive et un étudiant qui a attendu longtemps avant d'aller voir un cours ouvert dès le début.

D'autres données encore ne figurent pas dans le LMS car elles ne relèvent pas d'activités gérées par lui. Ainsi, dans bon nombre de formations, les données administratives de la formation ne sont pas directement gérées sur le LMS, non plus que les historiques de contact et de communication entre les étudiants, les professeurs et le personnel administratif qui ne seraient pas passés par la plate-forme. Or, ces contacts peuvent contenir la justification de certaines données : par exemple, un étudiant n'a pas pu se connecter de toute la semaine parce qu'il était malade ou en déplacement.

Dans le cadre de formations mixtes, toutes les données relatives aux formations présentielles n'y figurent pas non plus. De même si la formation impose un examen final en présentiel. Toutes ces données sont généralement conservées par les responsables de formation et leur équipe administrative ainsi que par les divers intervenants et enseignants, parfois dans des solutions peu adaptées comme des tableurs Excel.

L'étude de ces données présente toutefois autant d'intérêt que celle des données présentes sur le LMS, puisqu'elles permettent de dévoiler d'autres aspects de l'activité d'un étudiant ou de corriger les données approximatives fournies par le LMS. Notamment, on peut inclure dans les aspects non cernés par le LMS celui du contact humain et de l'évaluation humaine, certes subjective, mais néanmoins révélatrice d'une impression laissée. Il nous semble donc très important, dans la mesure du possible, d'inclure ces données dans notre projet de suivi des étudiants au même titre que les données plus faciles d'accès et de traitement.

Nous souhaitons donc ajouter à la liste de données brutes les données suivantes :

- le moment réel de fin de session
- la durée entre l'ouverture du cours et la première visite de l'étudiant
- la présence aux présentiels
- la note aux examens finaux
- l'historique et le contenu des contacts entre le personnel administratif et l'étudiant
- l'appréciation des différents enseignants et du responsable de formation

Maintenant que nous avons fait le tour des sources de données que nous souhaitons utiliser, nous allons maintenant présenter comment de telles données peuvent être utilisées pour construire des indicateurs.

2.2 Agrégation des données en indicateurs

Nous allons maintenant chercher à agréger ces données brutes en une liste d'indicateurs qui nous serviront au suivi des étudiants. Pour cela, nous allons d'abord chercher à comprendre mieux le processus de définition de ces indicateurs via l'état de l'art du domaine. Suivra une liste des indicateurs que nous proposons pour notre problématique, que nous comparerons à ceux proposés par des travaux similaires aux nôtres.

2.2.1 Indicateurs et traces d'interaction modélisées

Pour tenter de mieux comprendre ce qui se passe lors d'un processus d'apprentissage, et pour évaluer plus efficacement ces situations d'apprentissage, de nombreux chercheurs appliquent le même processus, celui d'analyser les traces d'interaction pour en tirer toutes sortes d'indicateurs. En effet, l'analyse des traces permet le contrôle et la régulation de l'activité d'apprentissage ([Heraud et al., 2005]).

Settoui définit une trace comme « une séquence temporelle d'observés » [Settoui et al., 2006]. Prié [Prié, 2006] a défini la notion de traces premières comme la version modélisée des traces brutes, qui leur correspond directement, sans aucune transformation. À partir de ces traces premières, on peut calculer des indicateurs plus élaborés.

Selon [Djouad et al., 2009], « concevoir un indicateur consiste alors à modéliser explicitement la trace d'interaction permettant de le calculer, mettant en évidence les éléments d'observation à recueillir pour l'élaborer. Mettre en œuvre concrètement l'indicateur consiste à expliciter les transformations nécessaires à partir des traces d'interaction disponibles, considérées comme traces modélisées « premières » jusqu'à la formule de calcul de l'indicateur se fondant sur des observés de haut niveau. Un « chemin » de transformation est ainsi disponible et exploitable explicitement pour tout à la fois documenter et exécuter le calcul de l'indicateur. »

Si la trace est dépendante de la plate-forme où elle a été enregistrée, ce n'est pas le cas de l'indicateur, qui, lui, est générique et donc réutilisable. C'est tout l'intérêt du travail de modélisation de l'indicateur, mais donc aussi de la trace qui lui correspond.

Selon [Dimitracopoulou et al., 2006], un indicateur est une variable mathématique. Elle peut prendre des valeurs numériques, alphanumériques ou même graphiques. Elle peut être brute (sans unité définie), calibrée ou interprétée. Enfin, chaque indicateur peut dépendre d'autres variables ou d'autres indicateurs.

Selon [Djouad et al., 2009], un indicateur est défini par l'association d'une règle de calcul et d'un modèle de trace.

La modélisation des indicateurs offre l'avantage de la possibilité de leur standardisation et de leur réutilisation. Pour cela, il faut passer par un format commun d'échange. Le langage UTL [Choquet et al., 2007] a été proposé à ces fins. Son extension, UTL-CL [Pham Thi Ngoc et al., 2009], permet de décrire également le calcul des indicateurs. À partir de cela, on peut imaginer l'existence de bibliothèques d'indicateurs, indépendantes des plate-formes.

Une modélisation UML basée sur le méta-modèle CIM (Common Information

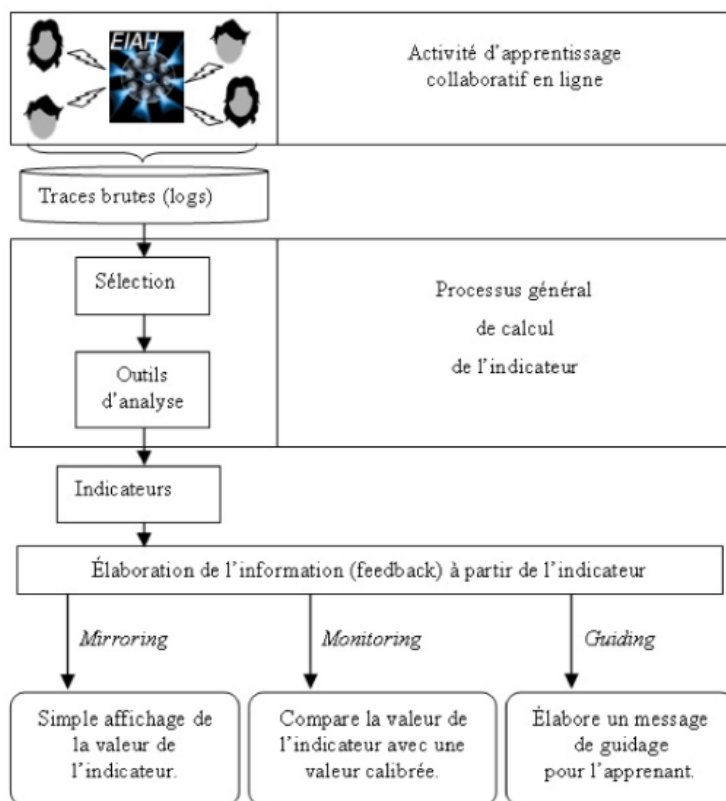


FIGURE 2.2 – Architecture d'un outil de calcul d'indicateurs

Model) a également été réalisée par [Broisin et al., 2007]. [Settoui et al., 2006] définissent un système à base de traces (SBT) comme un système informatique permettant et facilitant la manipulation des traces. [Diagne, 2008] avait également proposé une architecture permettant de réutiliser les indicateurs, mais ne prévoit pas de liens entre indicateurs, et ne prévoit pas non plus de formule de transformation des traces.

La figure 2.2¹ montre le fonctionnement d'un outil de calcul d'indicateurs. On voit que ces indicateurs peuvent avoir différents usages, dont celui du retour par affichage à certains utilisateurs, celui de l'évaluation de la valeur, voire de la comparaison avec d'autres valeurs ou seuils, et celui du guidage à partir de cette évaluation. Ces trois types d'usage ont été définis dans [Jermann et al., 2001].

Une première revue du domaine de la collecte et de l'agrégation d'information d'apprentissage a été proposée dans [Soller et al., 2005]. Ils y présentent les différents systèmes qui collectent les données d'interaction des étudiants.

1. Source : [Djouad et al., 2009]

2.2.2 Choix de nos indicateurs

À partir de ces données brutes, nous avons défini un certain nombre d'indicateurs permettant d'effectuer le suivi des étudiants. Nous avons cherché à en proposer avec une certaine diversité. Ainsi, nous avons cherché à la fois à pouvoir avoir une vue détaillée de l'activité d'un étudiant, et une vue globale. Nous avons également souhaité avoir à la fois des indicateurs relatifs et d'autres absolus.

Nous avons, préalablement à cette proposition, consulté des responsables de formation afin de leur demander quels indicateurs eux-mêmes utilisaient. Comme nous l'avions mentionné en introduction, ceux qui sont utilisés actuellement sont extrêmement restreints. Il s'agit du nombre de participations à quelques activités (quiz, devoirs, présentiels, autres exercices comme par exemple des mots fléchés) d'une part, et d'autre part de la date de dernière connexion et de la fréquence habituelle de connexion. Nous avons donc défini les indicateurs suivants en collaboration avec les responsables de formation afin d'améliorer la couverture des données mentionnées plus haut.

2.2.2.1 Indicateurs non normalisés

À partir des données brutes mentionnées plus haut, nous pouvons faire un certain nombre de traitements élémentaires qui nous donnent accès à des chiffres plus aisément manipulables qu'une énumération d'actions dans un historique. Ces traitements simples prennent la forme de statistiques, qui correspondent à des indicateurs non normalisés. Ils nous donnent accès à la liste de données suivantes :

- la durée de chaque session
- le moment de la journée auquel a lieu la session (matin, midi, soir, nuit)
- le moment de la semaine auquel a lieu la session (jours travaillés, week-end)
- la durée écoulée depuis la dernière connexion
- le nombre de sessions de connexion
- la quantité de leçons (sur le site et en PDF) et de ressources vues en tout
- la quantité de leçons (sur le site et en PDF) et de ressources distinctes vues (à comparer à la quantité totale disponible)
- la quantité d'activités, notées ou non, effectués en tout
- la quantité d'activités, notées ou non, distinctes effectuées (à comparer à la quantité totale disponible)
- la quantité de messages lus sur le forum
- la quantité de sujets créés sur le forum
- la quantité de réponses à un sujet existant écrites sur le forum
- la quantité de messages écrits sur le chat
- la note finale obtenue

Nous pouvons également tirer des données brutes mentionnées plus haut ainsi que des données traitées listée ci-dessus un certain nombre de statistiques qui nous semblent pertinentes pour refléter l'activité moyenne d'un étudiant. Nous trouvons notamment pertinent de mesurer :

- la durée moyenne des sessions

- la fréquence de connexion
- la note moyenne obtenue par l'élève aux activités notées
- la moyenne des notes obtenues par l'élève à sa dernière tentative aux activités notées, pour celles dont le contenu ne varie pas à chaque tentative
- la moyenne des meilleures notes obtenues par l'élève aux activités notées.

Ces deux catégories, les données traitées et les statistiques, offrent donc des grandeurs absolues, qui toutefois n'ont pas de valeur de référence universelle. En effet, d'une formation à l'autre, les nombres et même les ordres de grandeur en jeu peuvent varier complètement. Sur une formation d'une durée d'un an, une absence de connexion pendant une semaine n'est pas particulièrement inquiétante ; elle le sera davantage sur une formation de trois mois.

Bien que très basiques, il nous semble intéressant de noter qu'actuellement, ces données traitées et statistiques ne sont pas proposées par Moodle.

Nous pouvons comparer notre proposition à d'autres propositions existantes. Ainsi, dans [Gaudioso and Talavera, 2006], dont l'étude est centrée sur les interactions entre élèves, des statistiques assez proches sont étudiées. Toutefois, étant donné leur objet d'étude, il y a naturellement davantage d'indicateurs centrés sur l'utilisation du forum. Elles font comme nous la distinction entre consultation et écriture des messages ainsi qu'avec la création de nouveaux sujets, mais vont jusqu'à des détails que nous jugeons peu utiles, comme de savoir s'il y a eu ou non des réponses au sujet créé ou s'il est maintenant clos. Nous avons jugé que les indicateurs qu'elles proposaient sur la création de nouveaux forums et l'inscription à notre cours ne nous étaient pas applicables. En effet, les étudiants n'ont généralement pas de droits d'administration sur les forums, et notre étude cible des formations regroupant le même nombre de cours pour tous les élèves.

Dans [Romero et al., 2008], Romero et Ventura proposent des indicateurs assez basiques visant à démontrer la possibilité de les utiliser pour des fins d'apprentissage automatique. Ils rajoutent par rapport à nous des indicateurs de temps passé à effectuer diverses activités. Toutefois, ils ne précisent pas par quel résultat est obtenue la valeur de cet indicateur. C'est ce qui fait que nous n'avons pas souhaité le rajouter, car nous ne voyions que des moyens d'en obtenir des estimations peu précises, incapables par exemple de gérer le cas où un étudiant aurait plusieurs onglets ouverts et où des logs liés à diverses activités se mélangeraient dans la base de données.

Nous résumons la différence entre nos indicateurs et ces autres propositions dans un tableau, que nous avons dû découper en deux à cause de sa trop grande taille, et qui est présenté dans les figures 2.1 et 2.2. Ces données sont ensuite résumées dans le tableau 2.3. En le consultant, on peut vérifier qu'un grand nombre de nos indicateurs peuvent être considérés comme réellement innovants, puisque 15 d'entre eux ne figurent dans aucune des deux autres propositions, et que nous n'oublions rien de capital dans la mesure où nous n'avons laissé de côté aucun indicateur sur lequel les autres seraient d'accord entre eux.

Un autre jeu de statistiques a été utilisé par [Anozie and Junker, 2006] dans le contexte de l'étude de la prédiction de résultats d'élève de huitième année accompagnés par un système de tuteur intelligent nommé ASSISTment. Toutefois, les

Indicateur	Bovo et al.	Gaudioso et al.	Romero et al.
Nombre de sessions	oui	oui	non
Durée moyenne des sessions	oui	non	non
Fréquence de connexion	oui	non	non
Quantité de leçons lues	oui	nombre de pages statiques vues	non
Quantité de leçons téléchargées en PDF	oui	non	non
Quantité de leçons distinctes vues	oui	non	non
Quantité de leçons distinctes téléchargées en PDF	oui	non	non
Quantité de ressources vues	oui	nombre de pages statiques vues	non
Quantité de ressources distinctes vues	oui	non	non
Quantité d'activités non notées réalisées	oui	non	non
Quantité d'activités non notées distinctes réalisées	oui	non	non
Quantité d'activités notées réalisées	oui	non	quiz
Quantité d'activités notées distinctes réalisées	oui	non	devoirs seulement
Nombre de quiz réussis	non	non	oui
Note de quiz ratés	non	non	oui
Temps passé sur les quiz	non	non	oui
Temps passé sur les devoirs	non	non	oui

TABLE 2.1 – Comparaison des indicateurs non normalisés proposés par divers travaux

Indicateur	Bovo et al.	Gaudioso et al.	Romero et al.
Moyenne de toutes les notes	oui	non	non
Moyenne des dernières notes	oui	non	non
Moyenne des meilleures notes	oui	non	non
Messages lus sur le forum	oui	oui	oui
Messages écrits sur le forum	oui	oui	oui
Sujets créés sur le forum	oui	oui	non
Messages écrits sur le chat	oui	oui	oui
Nombre d'entrées sur le chat	non	oui	non
Activation des alertes du forum	non	oui	non
Création d'un forum	NA	oui	NA
Sujets créés ayant des réponses sur le forum	non	oui	non
Sujets créés et clos sur le forum	non	oui	non
Messages privés envoyés à l'enseignant	non	non	oui
Envoi d'un e-mail à tous les autres participants	non	oui	non
Temps passé sur le forum	non	non	oui
Nombre de fichiers dans l'espace de stockage	non	oui	non
Publication d'une présentation	non	oui	non
Ajout de marque-pages	non	oui	non
Inscription à d'autres cours	NA	oui	NA
Note finale obtenue	oui	non	oui

TABLE 2.2 – Comparaison des indicateurs non normalisés proposés par divers travaux (suite)

Bovo	Gaudioso & Talavera	Romero & Ventura	nombre d'indicateurs
oui	oui	oui	3
oui	non	non	15
oui	oui	non	2
oui	non	oui	1
non	oui	non	8
non	non	oui	6
non	oui	oui	0

TABLE 2.3 – Synthèse de la comparaison des indicateurs non normalisés proposés par divers travaux

indicateurs proposés sont tous extrêmement spécifiques à la plate-forme employée, et ne sont donc pas adaptables à notre étude.

D'autres jeux d'indicateurs peuvent correspondre à d'autres domaines. Ainsi, [Merceron et al., 2004] proposent également une liste d'indicateurs, mais ceux-ci sont plus adaptés à leur problématique de réingénierie pédagogique. [Barros and Verdejo, 2000] proposent des indicateurs adaptés à un travail collaboratif. Ceux de [Reffay et al., 2006] considèrent les forums de discussion comme des réseaux sociaux, et ceux de [May et al., 2007, Lavallard, 2008] permettent d'analyser l'activité d'un forum, et sont associés à des visualisations spécifiques.

2.2.2.2 Indicateurs normalisés

Il nous paraît également intéressant de ramener ces indicateurs à un nombre relatif permettant de comparer les étudiants entre eux plus aisément sans avoir en tête les valeurs caractéristiques pour chaque indicateur et chaque formation. Pour ce faire, nous allons avoir recours à un processus de normalisation.

À l'issue du processus de normalisation, nous souhaitons obtenir une note sur 10, qui nous semble être un format assez lisible et parlant. Nous avons choisi, pour les données basiques dont nous parlons actuellement, de dire que la note de 10/10 serait attribuée au meilleur élève du groupe sélectionné pour l'indicateur étudié. La note des autres élèves sera mise à l'échelle en proportion. Nous n'avons ainsi pas besoin d'effectuer d'autres normalisations car nous partons de l'hypothèse que le groupe d'élèves sélectionné est homogène : ils sont sur la plate-forme depuis la même durée et ont les mêmes activités à faire. Il ne serait pas pertinent de faire un calcul comparatif sinon.

Il faut toutefois noter que cette normalisation, si elle permet de situer les élèves au sein d'un groupe, ne permet pas de juger du niveau global de ce groupe.

Nous avons ainsi recensé un certain nombre d'indicateurs normalisés, que nous nommons également facteurs d'apprentissage :

- fréquence de connexion
- durée depuis la dernière connexion
- temps passé en ligne
- lecture de leçons en ligne
- lecture de leçons en PDF
- consultation de ressources
- participation au wiki
- participation au journal
- participation au blog
- quiz effectués
- devoirs effectués
- SCORM effectués
- Hot Potatoes effectués
- notes obtenues
- dernières notes obtenues

- meilleures notes obtenues
- lecture du forum
- création d'un sujet sur le forum
- réponse à un sujet sur le forum
- messages sur le chat
- participation à des visioconférences
- avis du tuteur

Nous pensons, à travers ces indicateurs, avoir réussi à capturer l'essentiel des divers aspects de l'activité d'un étudiant. Nous y voyons apparaître plusieurs grands thèmes : la présence en ligne, l'étude des contenus, la participation aux activités notées ou non notées, la participation sociale, qui relèvent tous d'une quantité d'activité ; et les notes obtenues, qui permettent de vérifier que cette activité n'est pas en vain.

Nous n'avons pas pour l'instant trouvé d'autres travaux jugeant utile de proposer une version normalisée de leurs indicateurs. Il est vrai que cette normalisation peut paraître être un traitement très simple, qu'il n'est pas la peine de mentionner. Toutefois, nous pensons que ces indicateurs ont un vrai rôle à jouer dans la compréhension aisée du comportement des étudiants, de par leur format extrêmement compréhensible et par la vision de groupe, contextualisée, qu'ils offrent.

Ces divers indicateurs ne sont manifestement pas indépendants. Notamment, ils dépendent tous de la valeur des indicateurs de présence, puisqu'un étudiant rarement connecté n'a pas de temps en ligne à consacrer aux autres activités.

Nous allons maintenant proposer une formule de calcul pour chacun de ces indicateurs, pour obtenir le nombre à normaliser.

Fréquence de connexion On part du nombre de connexions divisé par le nombre de jours de la période sélectionnée. On majore à 1 pour obtenir une estimation de la proportion de jours durant lesquels il y a eu une connexion, et ne pas être trop influencé par de multiples connexions dans une même journée. Soit :

$$\text{proportion} \text{ JoursConnecté} = \min(\text{nbConnect} / \text{nbJours}, 1)$$

Durée depuis la dernière connexion On retourne

$$\text{durée} \text{ DernièreConnexion} = \exp(-2 * \text{nbJours})$$

nbJours vaut le nombre de jours écoulés depuis la dernière connexion sur la période. Si l'élève ne s'est encore jamais connecté, $\text{nbJours} = +\infty$ et on retourne donc 0.

Temps passé en ligne Si c'est possible, on calcule la durée réelle de chaque session. Faute de mieux, on fait une estimation de type borne haute du temps passé par un élève sur la plate-forme, en connaissant le moment de connexion et soit le moment de déconnexion, soit le timeout de la plate-forme en son absence (si l'élève a quitté la page sans cliquer sur déconnexion). On additionne la durée de toutes les sessions enregistrées.

```
durée_session:
  si il existe une déconnexion de l'élève avant sa prochaine connexion:
    return date_déconnexion - date_connexion
  sinon:
    return timeout

temps passé en ligne:
  tps = 0
  pour toutes les connexions de l'élève:
    tps += durée_session
  return tps
```

Lecture de leçons en ligne, en PDF, Consultation de ressources, Participation au wiki, au blog, au journal, Quiz, SCORM, Devoir, Hot Potatoes, Lecture de forum Pour toutes ces activités, nous essayons de prendre en compte à la fois le nombre total et le nombre distinct. Il est crucial de pouvoir faire la différence entre quelqu'un qui a lu 5 leçons et quelqu'un qui a lu 5 fois la même leçon. Toutefois, il est quand même important de savoir que quelqu'un est venu relire une leçon pour se rafraîchir la mémoire sur un détail. Pour conserver un nombre raisonnable d'indicateurs, nous avons choisi d'intégrer ces deux éléments dans un seul calcul pondéré en faveur du nombre distinct :

$$note = (nbDistinct + (nbTotal/10))$$

avec :

- *nbDistinct* le nombre d'activités de ce type faites sur la période, chaque activité étant comptée une seule fois
- *nbTotal* le nombre d'activités de ce type faites sur la période, sans restriction.

C'est ce qui permet de faire la différence entre quelqu'un qui a fait 10 fois le même quiz et quelqu'un qui a fait 10 quiz différents une fois chacun.

Nouveaux sujets de forum, Réponses sur le forum, Messages sur le chat, Participation à des visioconférences Pour ces actions, la distinction entre nombre total et nombre distinct n'a pas de pertinence. Nous utilisons donc une formule simplifiée :

$$note = nbTotal$$

Moyenne de toutes les notes Comme son nom l'indique, on calcule la moyenne de toutes les notes obtenues par l'élève aux diverses activités notées, en les mettant au même dénominateur si ce n'était pas le cas. Aucune pondération n'est appliquée.

Moyenne des dernières notes Pareil, mais avec uniquement la dernière note obtenue à chaque activité.

Moyenne des meilleures notes Pareil, mais avec uniquement la meilleure note obtenue à chaque activité.

Avis du tuteur Le tuteur propose directement une note sur 10 reflétant son avis sur l'élève.

Nous avons présenté cette liste d'indicateurs dans les articles [Bovo et al., 2013b, Bovo et al., 2013c].

2.2.2.3 Indices thématiques et IGA

Nous avons mentionné à propos des indicateurs que nous venons de proposer qu'ils s'agrègent naturellement en grands thèmes. Il nous semble pertinent d'utiliser ces thèmes pour proposer de nouveaux indicateurs à un niveau de granularité plus élevé, pour avoir une vue plus globale de l'activité de l'étudiant. Ces indicateurs prennent également la forme d'une note sur 10, pour une plus grande simplicité de compréhension. Nous les appelons indices thématiques.

Les thèmes que nous avons vu se dégager et pour lesquels nous souhaitons proposer des indicateurs sont les suivants :

- la présence en ligne : à quelle fréquence l'étudiant se connecte-t-il et combien de temps passe-t-il connecté ?
- l'étude des contenus : l'étudiant accède-t-il aux leçons, ressources, etc. et pendant combien de temps ?
- la participation aux activités non évaluées : l'étudiant vient-il aux présentiels ? contribue-t-il aux wikis, blogs, journaux, sondages... ?
- la participation aux activités évaluées : l'étudiant fait-il les différents exercices, quiz, devoirs ?
- la participation sociale : l'étudiant se sert-il des forums ? Pour les lire, y écrire ? Obtenir de l'aide ou en fournir aux autres ? Participe-t-il au chat ?
- les résultats obtenus : obtient-il des bonnes notes à ces exercices, à ses examens ?
- les avis des tuteurs : de par son rôle privilégié de dialogue avec l'étudiant, le tuteur peut se forger une bonne idée de sa situation et prendre en compte des exceptions aux règles établies par le calcul précédent. Cet avis doit naturellement être pris en compte afin d'éviter de relancer un étudiant pour rien.

Le calcul de ces indices thématiques sera obtenu par une moyenne pondérée des indicateurs associés.

De même, afin d'avoir une vue globale la plus complète possible, nous souhaiterions donner à chaque élève une unique note sur 10 représentative de la quantité et de la qualité de son activité d'apprentissage. Nous avons baptisé cette note l'Indice général d'apprentissage (IGA). Elle sera une moyenne pondérée des indices thématiques.

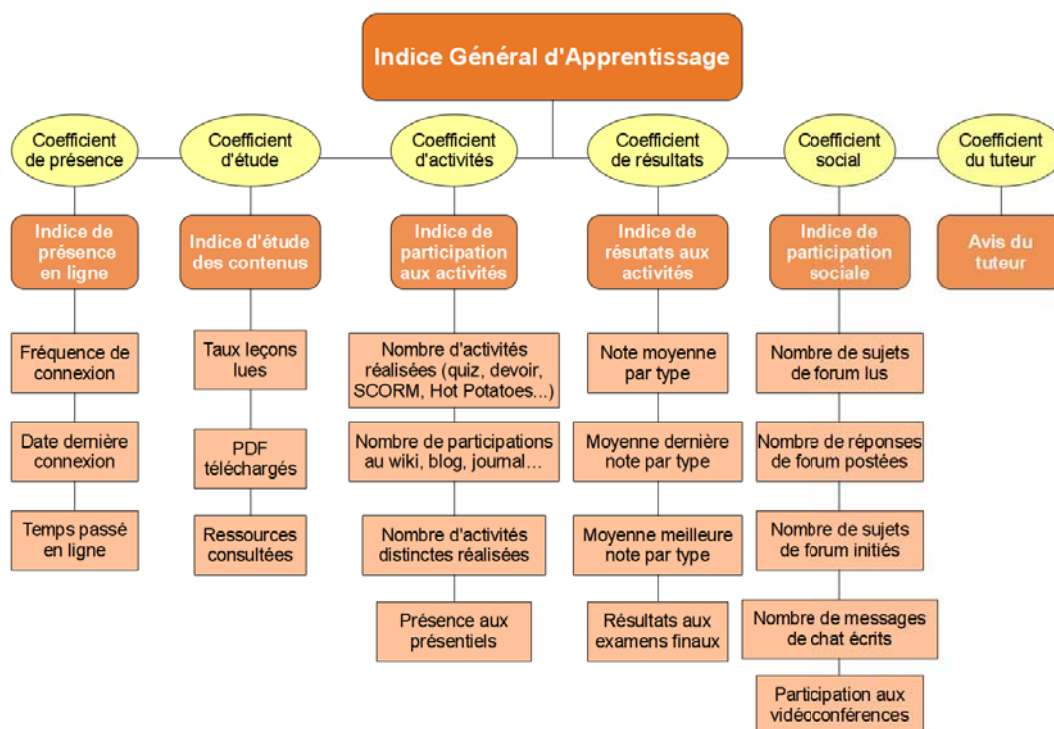


FIGURE 2.3 – Les indicateurs proposés

Il est à noter que ces calculs ne subissent pas de nouvelle normalisation après la moyenne pondérée. On peut donc s'attendre que pour ces indicateurs, personne n'obtienne la note maximale de 10, sauf s'il a été le meilleur du groupe pour tous les indicateurs qui vont être moyennés.

Nous proposons donc en tout trois niveaux hiérarchiques d'indicateurs, nous permettant d'observer l'activité d'un élève à différents niveaux de granularité. Au plus haut niveau, nous avons une seule note globale, appelée IGA pour Indice général d'apprentissage. Au niveau intermédiaire, nous proposons des indices thématiques qui permettent de se concentrer sur un seul aspect de l'activité de l'étudiant. L'IGA est une moyenne pondérée de ces indices thématiques. Au niveau le plus bas, chaque indice correspond à un certain type d'action de la part de l'élève, comme par exemple « répondre à un QCM ». Les indices thématiques sont une moyenne pondérée de ces indices, et l'IGA est une moyenne pondérée des indices thématiques. Tous ces indicateurs sont paramétrés par la sélection d'un élève, d'un groupe d'élèves à qui le comparer, et d'une fourchette de dates. Généralement, le groupe d'élèves choisi sera l'ensemble des élèves suivant une même formation. Au final, notre proposition est résumée dans la figure 2.3.

Nous avons fait le choix de ne pas proposer de pondération différente de la pondération par défaut (tous les indicateurs sont considérés comme ayant un poids égal). En effet, nous avons jugé que tous les indicateurs reflétaient une part différente et donc importante de l'activité d'un élève. Toutefois, notre proposition est adaptable et toute personne souhaitant s'en servir peut proposer une nouvelle pondération tenant compte de ce qui lui semble plus important.

2.2.2.4 Discussion

Il est très important de comprendre que ces indicateurs normalisés n'ont qu'une valeur relative. Cela peut être un piège dans lequel on tombe facilement en faisant la comparaison avec des notes normales. Ici, quelqu'un ayant une note de 2/10 n'est pas nécessairement mauvais dans le domaine en question. Il est peut-être comparé à quelqu'un qui a eu un taux d'activité exceptionnellement haut et qui tire donc les notes de tous les autres vers le bas. Nous avons eu ce cas dans une formation comprenant une élève étrangère, qui avait besoin de beaucoup plus étudier que les autres.

Pour éviter de tomber dans cet écueil, nous proposons divers outils.

Chacun des indicateurs que nous avons proposés visait à refléter le niveau relatif d'un élève par rapport à un groupe. Pour chacun d'entre eux, la valeur moyenne et la valeur médiane pour la formation pourront également être affichées, ce qui permettra de mieux situer l'apprenant au sein de ses camarades.

Pour chacun des indicateurs, on pourra également fixer des seuils qui correspondent à des valeurs absolues d'activité et indiquent à quelle note ces valeurs correspondent, ce qui permettra de mieux situer les élèves par rapport aux valeurs absolues.

Pour situer les élèves par rapport aux valeurs absolues, le mieux est d'utiliser les données brutes, données traitées et statistiques que nous avons mentionnées plus haut. Cependant, étant tous sur des échelles différentes, cela se prête moins à la lecture globale que nous avons voulu apporter ici. On pourrait imaginer une normalisation différente basée sur des seuils d'activités jugés convenables par le responsable de formation, mais ces seuils seraient à la fois différents pour chaque formation et, dans une certaine mesure, pifométriques. Nous n'avons donc pas souhaité développer cette approche.

Conclusion En résumé, les indicateurs que nous avons choisis sont basés sur plusieurs critères : les données que nous avons jugé pertinentes, la facilité de compréhension mais aussi la couverture la plus complète possible de l'intégralité des activités d'un apprenant. Ils reprennent souvent nos statistiques et les convertissent en une note.

Nous avons choisi de nous baser sur les données proposées par Moodle, pour ce qui relève des traces enregistrées par le LMS, mais nous considérons que les principaux LMS sont suffisamment semblables entre eux sur ce point pour considérer nos indicateurs comme génériques et applicables à n'importe quelle formation mixte ou entièrement sur LMS.

On peut imaginer plusieurs usages à ces indicateurs. En comparant les IGA sur plusieurs périodes, on pourrait éventuellement observer un taux de progression relative des étudiants au fil de l'année. À la fin de la formation, les IGA des utilisateurs pourront être confrontés avec leur taux de satisfaction et leur résultat aux examens afin d'affiner l'outil et de fournir un retour plus poussé aux responsables de formation.

Outre leur usage direct de consultation par les responsables du suivi des apprenants, ces indicateurs peuvent également servir de base à des études plus poussées. Ainsi, le chapitre suivant présentera des méthodes d'apprentissage automatique destinées à fournir davantage d'indications sur le comportement des étudiants. Nous verrons que ces méthodes d'apprentissage automatique sont basées sur des caractéristiques normalisées, rôle qui peut tout à fait être rempli par nos indicateurs normalisés.

Chapitre 3

Suivi par apprentissage automatique

Les indicateurs que nous avons présenté au chapitre précédent sont une première façon d'exploiter nos données pour mieux comprendre le comportement des étudiants et pouvoir ainsi mieux les suivre. Toutefois, il s'agit essentiellement de statistiques, normalisées ou non, bref, de traitements très simples appliquées à nos données. Des traitements plus complexes pourraient probablement nous apporter davantage d'informations et ainsi permettre d'améliorer encore le suivi des élèves.

Afin de trouver de telles méthodes, nous nous sommes tournés vers les domaines voisins de l'exploration de données et de l'apprentissage automatique. Ces domaines cherchent respectivement à extraire des connaissances structurées à partir de données brutes, et à modéliser des données par un processus d'apprentissage en l'absence de modèle plus explicite. Nous allons voir comment ces disciplines sont utilisées dans des domaines voisins du nôtre. Ensuite, nous allons prendre un peu de recul et faire un panorama plus complet de ces deux domaines, afin de voir quelles autres propositions d'utilisation de leurs méthodes pourraient nous servir.

Comme nous l'avons mentionné en conclusion du chapitre précédent, nos indicateurs normalisés pourront d'ailleurs être réutilisés comme entrées de certaines des méthodes que nous allons voir.

3.1 État de l'art

3.1.1 Introduction aux domaines de l'apprentissage automatique et de l'exploration de données

Dans un premier temps, nous allons commencer par présenter rapidement ces domaines et leurs problématiques, afin de mieux comprendre leur usage possible dans notre domaine.

3.1.1.1 Exploration de données

L'exploration de données est un domaine disciplinaire qui vise à extraire des connaissances structurées à partir d'une grande quantité de données non structurées par l'utilisation d'outils mathématiques et informatiques. Il cherche à construire un modèle sous-jacent aux données, qui permettra ensuite de pouvoir décrire plus facilement ces données et d'être utilisé pour en extraire des caractéristiques ou des prédictions. Selon [Fayyad et al., 1996a], les deux buts principaux de l'exploration de données sont la description, qui vise à rendre les données compréhensibles pour un humain, et la prédiction, qui consiste à prédire le comportement futur du système. On parle également à ce propos d'inférence statistique.

Il s'agit d'un champ à la frontière de plusieurs autres avec lesquels il est fortement relié : la statistique et l'intelligence artificielle, notamment l'apprentissage automatique, mais également la théorie de l'information. Il emploie des méthodes algorithmiques issues de ces domaines et y rajoute sa méthodologie.

Méthodologie L'exploration de données s'inscrit généralement dans un projet plus vaste de découverte de connaissance dans les bases de données (en anglais *KDD : knowledge discovery in databases*) qui comprend également la définition d'un but d'exploration, la collecte et la préparation des données, et la mise en forme et l'interprétation des résultats. Bien que ne faisant pas partie du domaine à proprement parler, ces étapes sont indispensables au bon déroulement du projet, puisqu'elles lui donnent son sens, et permettent d'éviter l'écueil parfois appelé drague de données (*data dredging* en anglais), dans lequel des motifs détectés par des algorithmes sont présentés sans analyse critique.

La découverte de connaissances dans les bases de données est généralement définie comme un processus de sélection, pré-traitement, échantillonnage et transformation de données à partir d'une base de données. Ce processus, illustré à la figure 3.1¹, peut avoir deux buts. Le premier, dit de découverte, est d'y découvrir des motifs, parmi lesquels en seront retenus un certain nombre, qualifiables de connaissances sur ces données. Le deuxième, dit de vérification, consiste à vérifier une hypothèse préalablement élaborée. L'étape d'évaluation des résultats est nécessairement un processus subjectif requérant une évaluation humaine [Fayyad et al., 1996b].

Dans ce cadre, l'exploration des données constitue une étape de ce processus, constituant en l'application de techniques et d'algorithmes produisant lesdits motifs à partir des données pré-traitées. Pour cela, on peut être amené à construire des modèles des données. Ces modèles jouent alors le rôle de connaissances inférées, dont l'intérêt sera évalué plus tard.

Le pré-traitement des données vise à remplir plusieurs rôles. Il décide que faire des éventuelles valeurs manquantes. Il peut permettre l'élimination des valeurs aberrantes ou extrêmes si cela est jugé nécessaire, mais également des transformations visant à réduire le nombre de dimensions en jeu, pour des raisons que nous évoquerons

1. Source : [Fayyad et al., 1996a]

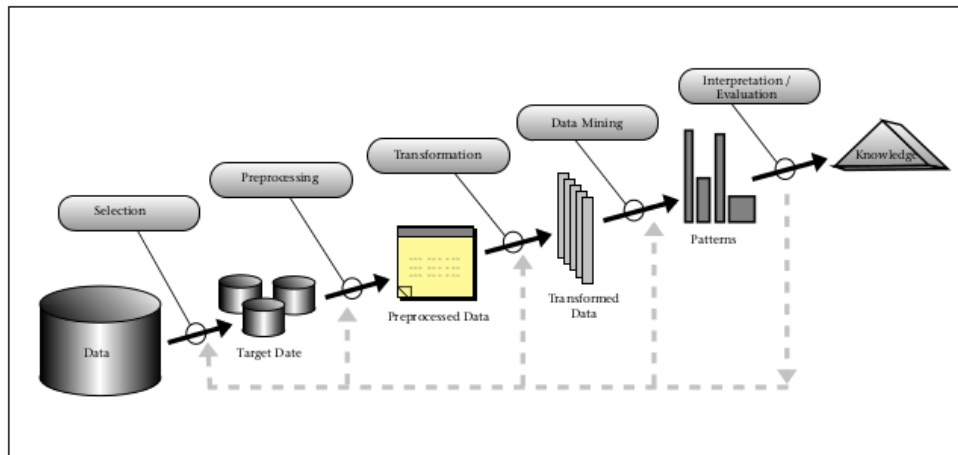


FIGURE 3.1 – Le processus d'exploration de données

plus tard. Enfin, un échantillonnage permettra de valider les résultats obtenus. Cet échantillonnage peut être spatial et/ou temporel.

On passe lors du processus de collecte, de sélection et de pré-traitement des données, d'une base de données qualifiée d'entrepôt de données, c'est-à-dire non spécifiquement préparée pour une étude, à un datamart, plus spécialisé.

De bonnes pratiques ont émergé au fil du temps pour améliorer la qualité des projets. Parmi celles-ci, les méthodologies aident les équipes à organiser les projets en processus. Au nombre des méthodes les plus utilisées se trouve la méthodologie CRISP-DM [Shearer, 2000], illustrée à la figure 3.2², qui découpe le processus de fouille de données en six étapes :

- la formalisation du problème posé
- la collecte des données, l'évaluation de leur qualité, la formation des premières hypothèses
- la préparation des données, en fonction des algorithmes qu'on voudra utiliser, qui peut nécessiter par exemple une discrétisation
- l'exploration des données (modélisation, segmentation, classification, régression, association...)
- l'évaluation des résultats et du processus, qui peut donner lieu à une nouvelle itération en cas d'insatisfaction
- la livraison et le bilan de fin de projet.

Avec ou sans l'application de règles méthodologiques, un certain nombre d'écueils sont possibles. Les plus courants ont été recensés par Robert Nisbet, John Elder et Gary Miner dans leur ouvrage *Handbook of Statistical Analysis & Data Mining Applications* [Nisbet et al., 2009] :

- utiliser des données peu fiables : les résultats ne peuvent pas être meilleurs

2. Illustration fournie par Kenneth Jensen : <http://sites.google.com/site/kennethagregardjensen/crisp-dm>

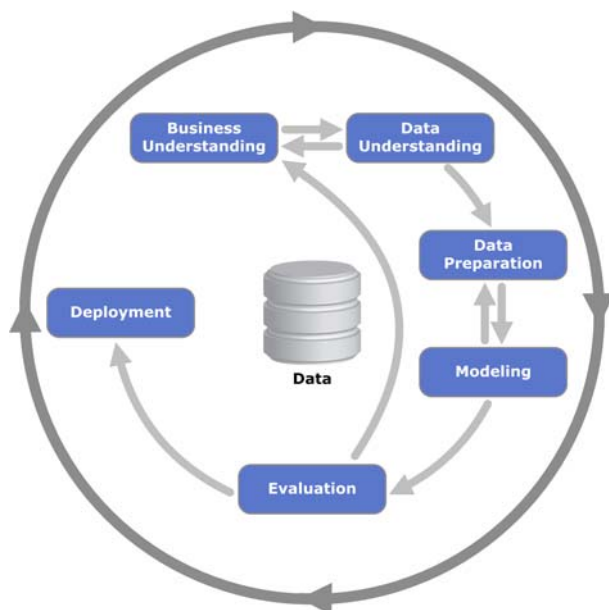


FIGURE 3.2 – La méthodologie CRISP-DM

que les données. Si l'on oublie de vérifier leur fiabilité, on peut en tirer des interprétations totalement fausses.

- poser la mauvaise question : cela mène à des résultats non pertinents.
 - recueillir trop peu de données par rapport à la complexité du problème : on ne pourra pas construire un modèle suffisamment complexe pour refléter fidèlement le problème.
 - mal construire son échantillon : cela mène soit à perdre en généralité, soit au contraire à un modèle trop imprécis.
 - n'utiliser qu'une seule technique algorithmique : cela ne permet pas la comparaison des résultats et risque de donner des mauvais résultats sur certaines données.
 - un sur- ou sous-apprentissage : cela fait qu'on ne peut pas généraliser les résultats en dehors de l'échantillon, et sera détaillé un peu plus bas.
 - ne pas remettre les données et les résultats dans leur contexte : cela mène à des erreurs d'interprétation.
 - éliminer des résultats inattendus comme paraissant aberrants : ces résultats peuvent être valides, il faut donc les vérifier.
 - extrapoler des résultats : cela mène quasi systématiquement à des erreurs.
- Selon George Box, « Tous les modèles sont faux, mais certains sont utiles » !

3.1.1.2 Apprentissage automatique

L'apprentissage automatique (*machine learning* en anglais) est un des champs d'étude de l'intelligence artificielle, qui essaie de mettre au point des algorithmes

correspondant à un processus d'apprentissage à partir de données fournies. Le but de ces algorithmes est très divers, et l'utilisation d'un processus d'apprentissage correspond à la prise de conscience que des méthodes plus classiques ne peuvent pas obtenir de suffisamment bons résultats dans certains domaines, qui ne peuvent pas être décrits de façon explicites. Ainsi, c'est notamment le cas de la reconnaissance d'écriture manuelle, qui impliquerait une description compliquée et fastidieuse de chaque lettre, devant prendre en compte les divers alphabets (dactylographié et manuscrit) ainsi que les variations d'une personne à l'autre : il est plus simple d'étiqueter un grand nombre d'exemples et de lancer un processus d'apprentissage à partir de ces données étiquetées.

La représentation des connaissances et des données est au cœur de ce domaine, tout comme elle l'est en exploration de données avec les modèles descriptifs. On cherche également à fournir un modèle capable de généralisation, c'est-à-dire capable de bonnes performances même sur des données nouvelles. Cette capacité de généralisation est la même que celle requise par les modèles prédictifs dans le domaine de l'exploration de données.

Tom M. Mitchell a proposé une formalisation de la définition de l'apprentissage par un programme informatique : « On dit qu'un programme informatique apprend d'une expérience E par rapport à un ensemble de tâches T et à une mesure de performance P si sa performance sur les tâches de T , telle que mesurée par P , s'améliore avec l'expérience E » [Mitchell, 1997].

Il existe un recoupement significatif entre l'exploration de données et l'apprentissage automatique. En effet, ces deux domaines utilisent souvent les mêmes méthodes et algorithmes. Toutefois, on peut les différencier par leur but, dans le sens où l'exploration de données est la seule à avoir un but de découverte de motifs dans les données. L'apprentissage automatique vise plus souvent l'exploitation de structures déjà connues. Ainsi, en apprentissage automatique, on comparera souvent les performances des algorithmes à des performances humaines sur des tâches réalisables par les deux, métrique qui n'aurait aucun sens en exploration de données. En exploration de données, les données étiquetées autorisant un apprentissage supervisé sont également beaucoup moins fréquentes.

3.1.2 L'exploration de données éducatives

L'idée d'utiliser les méthodes de la fouille de données et de l'apprentissage automatique dans le domaine de l'éducation est en plein essor depuis quelques années. Elle a donné naissance à une nouvelle sous-discipline, l'exploration de données éducatives (en anglais *Educational data mining*, ou EDM), dont quelques-uns des pionniers ont été Zaïane [Zaïane and Luo, 2001] et Beck et Woolf [Beck and Woolf, 2000].

Nous allons tenter ici de présenter quelques-unes des techniques utilisées dans cette discipline, du moins pour la partie qui s'intéresse à l'apprentissage en ligne, qui est le cadre de notre étude. Pour des revues plus détaillées, nous recommandons [Romero and Ventura, 2010, Romero and Ventura, 2007, Baker and Yacef, 2009]. Dans un premier temps, nous allons passer brièvement en revue les différentes techniques qui ne se rattachent pas à notre problématique du suivi des élèves, afin de donner une petite idée des champs d'exploration du domaine. Ensuite, nous nous attarderons davantage sur les techniques qui cherchent à comprendre la motivation et prédire la performance des élèves.

3.1.2.1 Panorama sur l'exploration de données éducatives

Nous allons lister ici brièvement quelques techniques utilisées dans le domaine de la fouille de données éducatives qui ne sont pas particulièrement rattachées à notre problématique. Le but est de donner une idée de la richesse et de la diversité de ce domaine d'études récent.

Construction de programme Il est toujours difficile de créer un programme d'éducation cohérent alors que divers savoirs peuvent s'enchevêtrer et qu'il faut décider dans quel ordre les étudier. Notamment, certains savoirs peuvent être des pré-requis à l'acquisition d'autres connaissances. Pour créer de meilleurs programmes ou tout simplement pouvoir les créer de façon automatique, plusieurs méthodes ont été proposées. Wang [Wang, 2005] a utilisé des algorithmes génétiques et présente pour cela un système d'évaluation de la qualité du programme. Vuong [Vuong et al., 2011] cherche à calculer les pré-requis en évaluant les performances des étudiants qui les ont préalablement suivis par rapport aux autres.

Analyse qualitative L'analyse qualitative s'intéresse à un événement précis et cherche à comprendre son contexte. Elle doit donc retrouver les données pertinentes autour de cet événement et permettre de les présenter sous forme compréhensible [Towne et al., 2002, Mostow et al., 2005]. Il s'agit ici d'essayer de mieux comprendre les résultats des élèves à une tâche donnée.

Recommandation La recommandation consiste à proposer à un élève de nouvelles tâches à faire, notamment de nouvelles ressources à lire. Elle permet de personnaliser l'apprentissage en offrant à chacun des ressources adaptées à son niveau actuel. Zaïane a été parmi les premiers à proposer un tel système avec l'apprentissage de règles d'association [Zaïane, 2002].

[Lu, 2004] propose une architecture commune pour les PLRS (Personalized Learning Recommender System).

[Tang and McCalla, 2003] utilisent une autre architecture qui utilise à la fois le regroupement des apprenants par leur centres d'intérêts avec le partitionnement, du filtrage collaboratif, et un module chargé de mettre à jour la liste de contenus à recommander.

[Zhang et al., 2008] cherchent d'abord à caractériser le style d'apprentissage des étudiants, puis utilise la fouille de motifs séquentiels avec l'algorithme AprioriAll pour déterminer leurs intérêts et leurs habitudes.

Analyse formelle de concept L'analyse formelle de concept (en anglais *Formal Concept Analysis*, FCA) s'attache à étudier les concepts dans un cadre qui précise leur contexte ainsi que les liens entre eux, c'est-à-dire à créer une ontologie. On peut définir un concept par extension, en listant l'ensemble des objets qui en relèvent, ou par intension, en listant tous les attributs que doivent posséder les objets qui en relèvent. Elle a été introduite par Rudolf Wille en 1982. On peut également parler de carte de concepts.

On peut faire un lien [Ganter et al., 2005] entre cette discipline et une forme de partitionnement de données, le biclustering, qui consiste à partitionner des objets sur la base de la valeur de certains attributs [Hartigan, 1972, Mirkin, 1998]. En effet, on peut considérer un concept comme un bicluster sur tous les attributs concernés.

Divers Diverses autres méthodes sont utilisées. Toutefois, nous n'avons pas repéré pour celles-ci une vraie fréquence d'utilisation en faisant un domaine d'étude. Nous les citons toutefois comme exemple pour montrer que le domaine ne se limite pas aux champs ci-dessus.

[Hoppe et al., 2003] utilisent un environnement d'apprentissage permettant de stocker les réponses des étudiants à ses exercices, ce qui lui permet ensuite de fouiller ces données pour en extraire les erreurs les plus fréquentes, les règles les plus souvent mal comprises, les erreurs souvent faites ensemble, les exercices les plus difficiles et l'avancement de chaque étudiant.

[Romero et al., 2003] utilisent un algorithme génétique pour découvrir des règles de prédiction sur les données issues d'un cours en ligne, et les comparent entre elles avec l'algorithme Apriori. Les règles découvertes montrent des liens entre certaines questions et soulignent la trop grande difficulté d'un exercice.

[Dringus and Ellis, 2005] proposent une liste d'indicateurs de l'activité d'un forum ainsi qu'une méthode pour en extraire automatiquement les valeurs.

[Feng et al., 2009] utilisent un modèle de régression logistique afin de comparer trois méthodes d'enseignement distinctes.

[Hurley and Weibelzahl, 2007] mettent en place un arbre de décision afin de décider dans quelles circonstances employer certaines interactions pédagogiques, et s'en servent pour créer une application visant à les recommander aux enseignants.

3.1.2.2 L'exploration de données éducatives centrée sur l'étude des apprenants

Nous allons ici présenter les diverses méthodes de fouille de données éducatives qui relèvent de l'étude des apprenants. Il va généralement s'agir de modéliser leur comportement et/ou leurs connaissances pour pouvoir prédire leur comportement futur et/ou leur performance.

Modèle cognitif Un modèle cognitif cherche à modéliser des processus psychologiques ou intellectuels humains afin de mieux les comprendre et de pouvoir en prédire les résultats. On parle généralement de modèle cognitif plutôt que d'architecture cognitive quand on se concentre sur la modélisation d'un seul processus ou sur l'interaction de deux d'entre eux, par opposition au processus plus complet cherchant à modéliser le plus fidèlement possible l'ensemble d'un agent. Ainsi, dans le domaine qui nous intéresse, on pourra chercher à modéliser le processus de l'apprentissage, ou la façon dont un élève résout un exercice. Cela peut amener à modéliser les diverses compétences requises pour résoudre un problème, et vérifier si l'élève les possède.

Le but de l'analyse des tâches est de lister et de décrire les tâches à effectuer pour accomplir un objectif. On cherche à analyser le comportement d'un humain amené à réaliser cet objectif afin de comprendre comment il le découpe en tâches, puis en actions associées. Si elle est bien réalisée, l'analyse de tâches peut permettre de prédire ou d'expliquer les performances d'un utilisateur dans un environnement donné. Lorsqu'on cherche à établir un modèle cognitif, on va généralement passer par une analyse de tâches avant d'associer les tâches et compétences trouvées aux problèmes et aux élèves. Cette analyse peut être un des sujets de l'apprentissage automatique. Ainsi, [Xu and Mostow, 2011] présentent une méthode intitulée LR-DBN (pour Logistic Regression in a Dynamic Bayes Net) pour modéliser l'effet des sous-compétences nécessaires à la réalisation d'une tâche.

Cette modélisation peut être manuelle ou automatique. Dans le premier cas, elle a généralement lieu par brainstorming et raffinements successifs du modèle, par des experts du domaine concerné et des experts en sciences cognitives. Ils vont chercher à capturer finement les différentes compétences présentes dans un programme d'apprentissage. Dans le deuxième cas, l'analyse en composantes principales peut quant à elle permettre de découvrir les compétences les plus pertinentes, auxquelles un professeur n'aurait pas forcément pensé. Dans les deux cas, on obtient ensuite un modèle probabiliste. On observe la réponse des élèves, et on cherche à modéliser leurs compétences.

Plusieurs raisons font qu'il est difficile de trouver un bon modèle cognitif :

- Même dans un domaine restreint, le nombre de compétences et de connaissances est généralement très important. On recense plus de 200 compétences dans le programme de mathématiques du collège.
- Parmi ces compétences, bon nombre d'entre elles ne sont jamais explicitées dans les manuels.
- Les compétences ne sont pas directement observables. On peut seulement les évaluer via des tâches qui y font appel.
- Les modèles manuels sont conçus par des experts, qui ont un point de vue très différent de celui d'un élève, tenant à tort certaines compétences pour implicitement acquises et ne voyant pas les difficultés au même endroit.

Ensuite, l'utilisation d'un algorithme d'apprentissage automatique supervisé permet de modéliser chaque élève comme ayant ou non les compétences modélisées. On peut utiliser divers algorithmes : régression, inférence bayésienne. . .

Parmi les premiers, [Beck and Woolf, 2000] ont utilisé plusieurs méthodes d'ap-

prentissage automatique (classifieurs bayésiens, arbres de décision, régression linéaire) pour créer un modèle d'étudiants capable de prédire leur performance à une question donnée ainsi que le temps qu'ils mettront à répondre.

L'intégration d'un tel modèle dans un tuteur virtuel intelligent peut fournir aux élèves des retours précis et des indices qui les aideront réellement à progresser, et peut sélectionner des problèmes à la difficulté adaptée à l'élève. Cette méthode, appelée l'enregistrement de connaissances (*knowledge tracing* en anglais), donne effectivement de meilleurs résultats de performance des élèves [Anderson et al., 1989].

Par exemple, Corbett et Bhatnagar [Corbett and Bhatnagar, 1997] ont utilisé le modèle de compétences d'Anderson, ACT-R [Anderson, 1993], pour déterminer par inférence bayésienne l'acquisition de ces règles.

Analyse des facteurs d'apprentissage Un modèle intitulé *Learning Factors Analysis* propose d'améliorer un modèle cognitif donné en combinant un modèle de régression logistique qui cherche à évaluer les compétences des élèves, l'intégration de données de difficulté dans la Q-matrice et une recherche combinatoire avec A* qui propose des modèles similaires et sélectionne le modèle le plus performant. [Cen et al., 2006] L'heuristique utilisée par A* est le critère d'information d'Akaike, ou le critère d'information bayésien.

Ce modèle a été ensuite réutilisé dans les travaux de Rafferty [Rafferty and Yudelton, 2007], qui propose de diminuer la complexité des modèles d'élèves utilisant de nombreux paramètres en utilisant des profils stéréotypés d'élèves. Ces profils sont basés sur le taux de succès à la première présentation du type de problème, et leur rapidité d'apprentissage. Cela permet d'améliorer les performances de la modélisation.

Le principal défaut de LFA est de supposer que la vitesse d'acquisition de tous les élèves est la même. Il a donc été adapté en un modèle plus adapté à la performance individuelle de chaque élève, PFA (*Performance Factors Analysis*) [Pavlik Jr et al., 2009b].

Ce modèle peut également être utilisé pour l'analyse de transfert de connaissances [Pavlik Jr et al., 2009a], qui modélise les relations entre les différentes compétences sous la forme d'une Q-matrice. Cette méthode appelée LiFT recherche des liens de type « implique nécessairement comme connaissance préalable » et est conceptuellement similaire à POKS [Desmarais et al., 1995] (partial order knowledge structure), qui recherche les liens inverses. Pour ce faire, on compare les vraisemblances des modèles dans lesquels les compétences sont reliées ou non.

Modèles de classification par diagnostic Les modèles de classification par diagnostic (*diagnostic classification models* en anglais) viennent du domaine des tests psychométriques. Ils cherchent à répondre aux insuffisances des tests classiques, qui ne proposent souvent comme résultat qu'une seule note, permettant uniquement de fournir un ordre entre les répondants au test. Ainsi, ils ne permettent pas de comparer qualitativement entre eux deux personnes ayant des notes similaires, ni de savoir si un élève a réellement maîtrisé une compétence donnée. Par leur approche

non orientée vers les compétences, ils ne permettent pas de fournir d'explication au score d'un répondant, et ne lui proposent donc pas non plus de piste pour améliorer ce score. Ils ne permettent pas non plus de savoir où mettre le seuil permettant de distinguer un succès d'un échec.

On cherche donc à modéliser les compétences d'un domaine donné, puis à évaluer les compétences de chaque personne. Pour cela, on procède à un diagnostic : on réalise un ensemble d'observations sous la forme d'un test, et on en déduit l'état sous-jacent qui a mené au résultat des observations. On conserve à l'issue du test non pas une note unique, mais un profil multidimensionnel, qui correspond à une liste de probabilités de maîtrise de chaque compétence.

Toutefois, il faut encore faire le lien entre les questions du test et les compétences qu'elles mesurent. Pour ce faire, on utilise une Q-matrice, qui permet d'indiquer ce lien [Anderson et al., 1989]. Cette matrice correspond également à celle utilisée en analyse confirmatoire de facteurs. Elle peut être calculée par des méthodes de types analyse en composantes principales.

On voit rapidement l'application que ces méthodes ont dans le domaine de l'éducation, où la question de la maîtrise des compétences est au cœur des problématiques des enseignants. Ces méthodes peuvent permettre de proposer des pistes d'amélioration aux élèves. Elles peuvent également permettre de mieux déterminer la note qui fait le seuil entre succès et échec.

Les deux algorithmes principaux de classification par diagnostic sont General Diagnostic Model (GDM) [Davier, 2008] et Loglinear Cognitive Diagnosis Model (LCDM) [Henson et al., 2009]. Le LCDM est similaire à un modèle ANOVA, avec deux différences : au lieu d'une sortie continue, le LCDM modélise la probabilité d'une réponse correcte ; et il est capable d'utiliser des variables cachées en plus des variables observables.

D'autres algorithmes sont centrés sur l'apprentissage de la Q-matrice [Tatsuoka, 1983, Liu et al., 2012a].

Analyse de parcours Il existe différentes approches d'analyse et de visualisation des données de navigation. Par exemple, Berendt et Brenstein ont proposé STRAT-DYN en 2001 [Berendt and Brenstein, 2001], qui propose trois groupements perceptuels distincts : l'ordre des visites des nœuds (qui souligne la découverte de l'information), les mouvements dans l'hypertexte (qui souligne la nature de la navigation) et la tâche expérimentale (représentation des nœuds et liens pour les différentes tâches expérimentales). On compare ensuite les comportements de navigation à des motifs connus : largeur d'abord, profondeur d'abord, points de repère... D'autres présentations de ces données ont été proposées par [McEneaney, 2001, Boechler and Dawson, 2002].

Amadiou, Bastien et Tricot [Amadiou et al., 2008] ont proposé diverses métriques statistiques d'évaluation de la recherche d'information à partir des parcours, comme par exemple le taux de précision, qui correspond au fait que l'utilisateur ne sélectionne que peu de documents non pertinents à sa tâche.

En étudiant le temps passé par les élèves sur chacune des pages de son envi-

ronnement d'apprentissage virtuel (VLE), David Monk s'est rendu compte que les élèves scannaient plus les pages à la recherche d'information qu'ils ne les lisaient attentivement, ce qui lui a permis de proposer des formats de ressources plus adaptés [Monk, 2005].

Novikoff s'est intéressé à l'ordre optimal de présentation des matériaux pour permettre aux élèves d'apprendre et de retenir le maximum de connaissances en un temps donné [Novikoff et al., 2012].

Partitionnement d'élèves en groupes de comportement similaire Dans le domaine de l'e-learning, le partitionnement peut notamment être utilisé pour constituer des groupes d'élèves de comportement similaire. Par exemple, un étudiant peut avoir trois types de comportement par rapport à l'utilisation d'un forum : il peut être actif en termes de lecture et d'écriture, actif seulement en termes de lecture ou ne les lire que rarement [Beaudoin, 2002, Taylor, 2002, Talavera and Gaudio, 2004, López et al., 2012]. Lawless et Brown [Lawless and Brown, 1997] distinguent trois profils de navigation : les chercheurs de connaissances, les explorateurs de caractéristiques et les utilisateurs apathiques.

[Tian et al., 2008] cherchent à constituer des groupes d'élève à la personnalité psychologique similaire, pour la confronter avec leur stratégie d'apprentissage. [Zakrzewska, 2008] préfère les regrouper suivant leur style d'apprentissage. On peut alors s'appuyer sur ces groupes pour personnaliser l'encadrement de l'élève.

Le partitionnement peut aussi aider à regrouper les élèves par centre d'intérêt pour leur recommander des ressources [Tang and McCalla, 2003].

[Tang and McCalla, 2002] proposent un algorithme de partitionnement basé sur les séquences fréquemment répétées pour repérer les élèves dont le comportement en termes de parcours des ressources est similaire.

[Ayers et al., 2009] utilisent le partitionnement à d'autres fins, pour comparer les compétences des élèves mesurées de différentes façons à partir d'une population simulée.

Détection de mauvais usages Des formes de classification peuvent servir à détecter des usages non désirés des plate-formes. Ainsi, [Baker et al., 2004a] ont réussi à identifier des comportements d'élèves démotivés qui cherchaient juste à obtenir le plus vite la validation permettant de passer à la suite, sans chercher à savoir quelle était la bonne réponse ni pourquoi. Toutefois, cette approche est principalement disponible sur les tuteurs virtuels intelligents, où les élèves peuvent demander beaucoup d'indices.

Apprentissage de règles d'association [Merceron and Yacef, 2008] utilisent l'apprentissage de règles d'association pour observer des tendances sur l'utilisation des ressources pédagogiques faite par les étudiants.

[Minaei-Bidgoli et al., 2004] proposent un algorithme pour repérer des règles de contraste, c'est-à-dire à observer les caractéristiques qui vont produire des résultats

différents en mettant en contraste deux règles produisant des résultats contraires. Parmi leurs résultats étonnants, ils trouvent que chez les bons élèves, les filles ont tendance à attendre plus longtemps que les garçons avant de retenter un exercice raté.

[Romero et al., 2009] utilisent un algorithme évolutionnaire pour apprendre des règles floues décrivant les causes des notes des étudiants en terme de leur activité préalable.

[Ma et al., 2000] cherchent à prédire quels élèves auront besoin de cours de soutien scolaire.

Prédiction de performance La classification est particulièrement utilisée pour prédire les performances d'un élève.

[Superby et al., 2006] cherchent à classier les étudiants en trois groupes suivant leur risque d'échec ou d'abandon durant leur première année universitaire, en utilisant les données d'un questionnaire posé durant cette année. Pour cela, ils utilisent l'analyse discriminante, des réseaux de neurones, des arbres de décision et des forêts d'arbres décisionnels, avec une performance toutefois qualifiée de « pas remarquable ».

[Herzog, 2006] emploie des arbres de décision et des réseaux de neurones pour chercher à prédire quels élèves seront diplômés au bout d'un certain nombre d'années, et quels élèves abandonneront.

[Dekker et al., 2009] emploient également des arbres de décision pour prédire les élèves qui vont abandonner leurs études, ce qui lui permet d'identifier les cours dont le résultat est le plus décisif.

[Barker et al., 2004] utilisent des réseaux de neurones et des machines à vecteur de support pour prédire combien de temps les élèves mettront à être diplômés.

[Baradwaj and Pal, 2012] utilisent un arbre de décision avec l'algorithme ID3 pour prédire la performance des élèves à partir de critères tels que la moyenne des notes, les notes au semestre précédent, le taux de présence et les notes aux diverses activités reliées au cours.

[Minaei-Bidgoli et al., 2003] utilisent une combinaison de plusieurs classifieurs, optimisée grâce à un algorithme génétique, pour prédire dans quelle gamme de valeurs sera la note finale des élèves à partir d'indicateurs assez semblables aux nôtres.

[Gedeon and Turner, 1993] utilisent un réseau de neurones pour prédire la moyenne finale des élèves à partir des notes en cours d'année, avant l'examen final, et en profitent pour essayer de fournir une explication au résultat, afin de fournir aux élèves de meilleures possibilités pour améliorer leurs résultats. [Calvo-Flores et al., 2006] utilisent également des réseaux de neurones à cette fin.

[López et al., 2012] utilisent le partitionnement comme base à la classification pour chercher à prédire le succès des élèves, et obtiennent les meilleurs résultats avec l'algorithme Expectation Maximisation.

Des travaux équivalents utilisent la classification pour estimer le succès des élèves à un seul cours [Zang and Lin, 2003], ou à une seule tâche [Desmarais and Pu, 2005].

Toutefois, la classification n'est pas la seule méthode utilisée pour la prédiction de performance, comme le montrent les exemples suivants.

[Pardos et al., 2007] utilisent un réseau bayésien pour prédire la performance des élèves et montrent au passage que les modèles les plus granulaires, utilisant le plus grand nombre de compétences des élèves en entrée, sont plus performants.

[Nebot et al., 2006] utilisent un système expert basé sur le raisonnement inductif flou et des règles d'association pour prédire, avec une bonne précision et assez peu de caractéristiques, la note finale des élèves.

[Anozie and Junker, 2006] utilisent un modèle de régression linéaire pour chaque mois de l'année scolaire pour étudier la variation de la prédiction de la note finale d'un élève.

Classification Nous venons de voir qu'un des premiers usages de la classification est de chercher à prédire le succès des élèves à leur formation. Toutefois, elle peut également être utilisée pour détecter le type de comportement des élèves. Ainsi, [Cocea and Weibelzahl, 2011] ont étudié les critères les plus révélateurs de la démotivation d'un élève par des méthodes de classification, et [Villaverde et al., 2006] ont utilisé des réseaux de neurones pour dévoiler le type d'apprentissage d'un élève.

3.1.2.3 Analyse

Nous avons présenté diverses méthodes d'utilisation des méthodes d'exploration de données et d'apprentissage automatique dans le domaine de l'apprentissage en ligne. Parmi ces méthodes, seules quelques-unes ont été utilisées pour répondre à des problématiques semblables à la nôtre.

La modélisation cognitive, bien que très intéressante dans la mesure où elle peut prédire la réussite des élèves, sort un peu du cadre de notre étude qui se limite davantage à l'analyse comportementale des élèves. Il en va de même pour les autres méthodes qui sont basées sur la modélisation des connaissances à acquérir au cours de l'apprentissage, comme la classification par diagnostic ou l'analyse de facteurs d'apprentissage. Bien que traduisant un aspect comportemental des élèves, l'analyse de parcours ne nous semble pas pouvoir refléter leur motivation, ni forcément être bien corrélée à leur performance.

En revanche, le partitionnement visant à créer des groupes d'élèves de comportement similaire nous paraît très intéressant. Nous pensons l'appliquer sur nos indicateurs afin d'observer les groupes formés. La classification nous permettra également de prédire le succès, l'échec ou l'abandon d'un élève en cours de formation. La régression nous permettra de prédire la note finale d'un élève. L'apprentissage de règles d'association nous paraît également très intéressant, puisqu'il nous permettra d'observer les corrélations entre nos indicateurs.

Nous allons maintenant prendre un peu de recul en étudiant les diverses possibilités offertes par l'exploration de données et par l'apprentissage automatique. Parmi elles, certaines pourraient n'être pas encore exploitées. De toute façon, ce panorama nous permettra de clarifier la nature des méthodes que nous venons de repérer.

Par la suite, nous élaborerons nos propositions d'utilisation des méthodes retenues dans la partie 4.3.

3.1.3 Panorama sur les méthodes de l'exploration de données et de l'apprentissage automatique

Nous allons maintenant plonger plus en profondeur dans la description des méthodes principales utilisées dans les domaines de l'exploration de données et de l'apprentissage automatique. Cette étude nous servira à mieux comprendre à quoi correspondent exactement les méthodes que nous venons de voir décrites dans la partie précédente, et à repérer si certaines méthodes non encore utilisées pourraient nous servir à proposer un meilleur suivi pour les apprenants.

3.1.3.1 Exploration de données

Diverses méthodes peuvent être utilisées dans le domaine de l'exploration de données. On peut les classer selon un critère : la modélisation est-elle à visée descriptive ou prédictive ?

Modélisation descriptive vs. prédictive Si les méthodes descriptives visent à mettre en évidence des informations présentes mais cachées par le volume des données, et donc de réduire, résumer et synthétiser les données, les méthodes prédictives cherchent davantage à expliquer les données par rapport à un ou plusieurs paramètres donnés, avec une variable cible dont on cherche à prédire la valeur. Nous allons comparer davantage ces deux méthodes de modélisation.

Méthodes descriptives Les méthodes descriptives permettent d'organiser, de simplifier et d'aider à comprendre l'information sous-jacente d'un ensemble important de données. Parmi ces données, aucune variable n'est considérée comme ayant davantage d'importance qu'une autre.

Parmi les informations qu'on peut chercher à obtenir figurent des regroupements d'individus en groupes homogènes ou la définition d'une norme par rapport à laquelle sera définie une anomalie. Parfois, la compression de données est l'effet désiré.

Pour obtenir ces résultats, on a accès à deux groupes de techniques. Parmi les techniques issues du domaine de la statistique, en plus des calculs de type moyenne, médiane, variance, écart-type, l'analyse factorielle permet de repérer des variables cachées du système. Ces variables, nommées facteurs, permettent souvent une nouvelle représentation plus parcimonieuse du système. Les algorithmes correspondant à l'analyse factorielle sont l'analyse en composantes principales, l'analyse en composantes indépendantes, l'analyse factorielle des correspondances, l'analyse des correspondances multiples et le positionnement multidimensionnel.

Si les données sont des séries temporelles, cela ouvre de nouvelles possibilités d'analyse, comme l'autocorrélation des données, l'estimation de tendances, la décomposition ou l'analyse spectrale (ou fréquentielle).

L'autre groupe de méthodes vient du domaine de l'apprentissage automatique.

Le partitionnement de données permet d'obtenir des groupes homogènes et différents entre eux, bien qu'inconnus par avance. On doit donc avoir une mesure de similarité des données, et les données doivent être similaires au sein d'un même groupe et plus éloignées des données des autres groupes. Les algorithmes correspondant sont les k-moyennes et ses variantes, certaines variantes de réseaux de neurones comme les cartes auto-organisatrices, l'algorithme Espérance-Maximisation et la classification hiérarchique. Ce partitionnement peut également être flou plutôt que déterministe, comme avec l'algorithme FCM (Fuzzy C-Means).

La recherche de règles d'associations cherche à établir des corrélations entre les données et entre les variables. La recherche de règles d'association peut être utilisée de manière supervisée ; les algorithmes APriori, GRI, Carma, méthode ARD ou encore PageRank se servent de cette technique.

Enfin, le résumé cherche à fournir une représentation plus compacte des données. Cette méthode est principalement utilisée avec des données textuelles.

Par la suite, l'utilisation de méthodes descriptives peut ensuite donner lieu à un certain nombre de visualisations, qui serviront lors de l'interprétation des résultats. Si la visualisation est un outil essentiel pour faciliter la compréhension par un humain, elle est parfois difficile à réaliser, si les données sont en plus de 3 dimensions. On peut alors utiliser des projections, mais qui s'accompagnent nécessairement de perte d'information.

Méthodes prédictives Les méthodes prédictives permettent de prévoir un ou plusieurs phénomènes observables et de pouvoir confronter leur prédiction à la réalité. On choisit une ou des variables privilégiées comme étant les cibles de la prédiction, les autres variables étant alors utilisées comme informations sur lesquelles la prédiction va s'appuyer.

On peut dire que la prédiction réussie est bien souvent signe d'une meilleure compréhension et modélisation du phénomène qu'une simple explication. C'est d'ailleurs un critère permettant de juger la valeur d'une théorie scientifique, comme l'a répondu fameusement Laplace à Napoléon qui le questionnait sur la pertinence de l'hypothèse de l'existence d'un dieu en science :

« Cette hypothèse, sire, explique en effet tout, mais ne permet de prédire rien. En tant que savant, je me dois de vous fournir des travaux permettant des prédictions. »

On peut chercher à faire trois types d'opérations. La classification ou discrimination cherche à séparer la population en plusieurs classes, comme le partitionnement que nous avons prévu plus haut, mais on s'intéresse ici à expliquer comment les valeurs de diverses variables influent sur ce partitionnement, de façon à pouvoir prédire le groupe d'un nouvel arrivant. La régression, quant à elle, cherche à prédire une valeur numérique d'une ou de plusieurs variables par rapport à toutes les autres, ce qui revient à établir une distribution conditionnelle. Quant à l'analyse prédictive, elle cherche à analyser le présent et le passé pour prédire le futur.

Là encore, on retrouve deux groupes de méthodes, suivant leur domaine de provenance. Le domaine de la statistique a fourni les diverses techniques de régression, l'analyse discriminante prédictive, pour prédire une variable catégorielle, ou bien le modèle linéaire généralisé (GLM), le modèle additif généralisé (GAM) ou le modèle log-linéaire afin de prédire une variable multidimensionnelle.

Pour la prévision, on modélise fréquemment les données sous forme de séries temporelles. Se rajoutent pour les analyser des méthodes basées sur des moyennes flottantes ; des méthodes autorégressives, où l'on cherche à décrire la série en fonction de son propre passé, et non en fonction d'autres variables ; et des modèles intégrés. Les modèles les plus connus combinent ces approches : ce sont ARMA (*Auto Regressive Moving Average*) [Box et al., 1976], ARIMA (*Auto Regressive Integrated Moving Average*) et sa généralisation à plusieurs variables VAR (*Vector Autoregression*) [Sims, 1990].

Du côté de l'intelligence artificielle, l'inférence bayésienne pourra permettre de calculer les probabilités de réalisation d'un événement, et des valeurs numériques pourront être obtenues à partir de réseaux de neurones. Les séparateurs à vaste marge (SVM) permettent de faire de la discrimination aussi bien que de la régression.

Les tâches impliquées En résumé, on retrouve six grandes classes de tâches qui caractérisent l'exploration de données, qui sont rappelées ci-dessous et classées dans le tableau 3.1 :

- La détection d'anomalie, qui vise à identifier des données inhabituelles ou des erreurs qui peuvent demander une analyse plus approfondie par un humain.
- L'apprentissage d'association, qui recherche des liens entre différentes variables, par exemple des produits fréquemment achetés ensemble.
- Le partitionnement, qui cherche à découvrir des groupes ou des structures jusqu'alors inconnus dans les données.
- La classification, qui cherche à appliquer une structure connue à de nouvelles données, par exemple pour décider si un e-mail est du spam ou non.
- La régression, qui tente de modéliser les données avec le moins d'erreur possible. On peut voir la prédiction comme une forme de régression particulière appliquée à une série temporelle.
- Le résumé, qui cherche à fournir une représentation plus compacte des données.

Nous avons parlé plus haut de l'apprentissage automatique, qui est une discipline à part entière. Nous allons maintenant en proposer une brève définition afin de mieux comprendre les enjeux et problèmes de ce domaine, que l'exploration de données nous amène à beaucoup utiliser.

	Méthodes descriptives		Méthodes prédictives	
		Série temporelle		Série temporelle
Méthodes issues des statistiques	Analyse factorielle ; moyenne, médiane, variance, écart-type	Autocorrélation, estimation de tendances, décomposition, analyse spectrale	Régression	Moyennes flottantes, autorégression, modèles intégrés, prévision
Méthodes issues de l'apprentissage automatique	Partitionnement ; apprentissage de règles d'association ; résumé ; détection d'anomalie		Classification ; régression	Prévision

TABLE 3.1 – Résumé des principales méthodes d'exploration de données et de leur discipline d'origine

3.1.3.2 Apprentissage automatique

Nous allons dans un premier temps étudier les critères suivant lesquels on peut classer les méthodes d'apprentissage automatique, qui correspondent au but recherché et aux types d'entrées fournies. Dans un deuxième temps, nous en listerons les méthodes, classées selon ces critères.

Comme nous l'avons déjà mentionné dans la rubrique dédiée à l'exploration de données, ces algorithmes se heurtent souvent à des complexités très grandes, soit qu'il s'agisse de la malédiction de la dimensionnalité, soit tout bonnement par explosion combinatoire lorsqu'on essaie de combiner un grand nombre de paramètres.

Degrés de supervision

Apprentissage supervisé L'apprentissage supervisé utilise des données étiquetées avec la sortie désirée. Plus formellement, on fournit au système un ensemble de couples de vecteurs (P, A) qui définissent pour chaque perception P la sortie attendue A . Ces informations fournies sont appelées la supervision. Le système cherche donc à imiter ce comportement, mais également à le généraliser à de futures entrées non étiquetées. On a donc une première phase d'apprentissage hors ligne et une deuxième phase de test en ligne. Ce modèle est également appelé analyse discriminante.

Parfois, plutôt qu'une étiquette correspondant à une classe unique, on préfère associer à une donnée une probabilité d'appartenance à chacune des classes possibles. On parle alors d'apprentissage supervisé probabiliste.

C'est ce type d'apprentissage qui est essentiellement soumis au sur-apprentissage.

Apprentissage semi-supervisé Une forme d'apprentissage voisine est l'apprentissage semi-supervisé, qui combine des exemples étiquetés et non étiquetés durant la phase d'apprentissage. C'est essentiellement utile quand il est difficile ou coûteux d'obtenir des données étiquetées. Les données non étiquetées peuvent servir à apprendre la distribution des données, ce qui permet ensuite de mieux les classifier. L'apprentissage semi-supervisé exploite donc le partage statistique entre l'apprentissage des distributions de probabilité a priori et l'apprentissage des distributions de probabilités conditionnelles.

On peut comparer cet apprentissage à celui fait par les enfants en bas âge : par exemple, un enfant voit de nombreux exemples de chiens, et sur quelques-uns d'entre eux, entend les adultes autour de lui prononcer le mot « chien ». Il apprend alors à généraliser l'usage de ce terme à tous les chiens.

Apprentissage par renforcement L'apprentissage par renforcement diffère de l'apprentissage supervisé classique parce qu'il n'offre pas d'étiquetage a priori des données, mais propose a posteriori une évaluation de la sortie proposée par le système. Cette évaluation est ensuite utilisée par le système comme outil d'adaptation afin de chercher à maximiser la récompense obtenue.

Plus formellement, on fournit au système une perception P , pour laquelle il calcule sa sortie S . En comparant la sortie S à notre sortie attendue A , on calcule une récompense R que le système cherche à maximiser.

Ce type d'apprentissage est très bio-inspiré, puisqu'on voit directement l'analogie avec le comportement d'un être vivant dans un environnement complexe, pour lequel on ne peut pas directement connaître les conséquences d'une séquence d'actions avant de l'avoir réalisée. Ici, l'apprentissage sert donc à mieux comprendre cet environnement complexe et à s'y adapter.

Pour réussir au mieux, il faut s'assurer que le système est capable d'explorer au maximum l'environnement [Sutton and Barto, 1998]. En effet, le signal de renforcement indique à quel point, pour chaque état, une action est cohérente, mais en aucun cas si cette action est la plus pertinente de l'ensemble des actions possibles. Cela peut éviter le piège de tomber dans un maximum local.

Apprentissage non supervisé On parle d'apprentissage non supervisé quand l'ensemble des données non seulement n'est pas étiqueté, mais qu'aucune information supplémentaire à leur sujet n'est fournie. Le but de ce type d'apprentissage est de découvrir les structures présentes dans les données, par exemple avec le partitionnement de données. On retombe donc ici davantage dans la problématique de l'exploration de données. Cette méthode est souvent source de sérendipité.

En plus de ces grandes catégories, il existe quelques autres méthodes moins usitées. Par exemple, dans l'apprentissage actif, l'algorithme sélectionne les points de données sur lesquels il souhaiterait obtenir une supervision. Idéalement, le modèle doit donc sélectionner les données dont il pense pouvoir apprendre le plus.

En ligne vs. hors ligne L'apprentissage en ligne se fait en recevant les instances une par une, en même temps qu'une prédiction en est demandée. Idéalement, on reçoit rapidement une confirmation ou une infirmation de cette prédiction, de façon à avoir un retour sur l'apprentissage. Dans ce cadre, apprentissage et fonctionnement de l'algorithme ont donc non seulement lieu en même temps, mais sont intrinsèquement liés. Par contraste, un algorithme hors ligne reçoit dès le début l'intégralité des données d'apprentissage, et les traite aussitôt pour en construire un modèle complet.

Non seulement l'incomplétude du modèle peut provoquer des erreurs tôt dans la phase d'apprentissage d'un algorithme en ligne, mais il est également possible que les décisions prises tôt en présence de peu de données contraignent le modèle obtenu plus tard et le rendent non optimal.

Paresseux vs. hâtif L'apprentissage paresseux ne cherche pas à généraliser ses connaissances avant qu'une requête le nécessitant lui soit faite. Par opposition à la méthode classique, dite hâtive, cela peut permettre de faire de très bonnes approximations locales ; en revanche, cela nécessite un espace mémoire suffisant pour stocker l'intégralité des données d'apprentissage, ce qui est une contrainte forte. Si l'apprentissage est plus rapide, l'évaluation d'une nouvelle donnée est plus lente.

Modèle génératif vs. discriminatif vs. prédictif Les modèles discriminatifs visent essentiellement à apprendre une loi de probabilité conditionnelle. Ces modèles sont nécessairement produits par apprentissage supervisé et ne peuvent pas être utilisés pour générer des données. Ils servent essentiellement à des tâches de classification et de régression.

Au contraire, les modèles génératifs s'attachent à pouvoir générer des données, par exemple à des fins de simulation, et doivent donc connaître non seulement les probabilités conditionnelles, mais également les distributions de probabilité de toutes les variables concernées.

Enfin, un modèle prédictif vise à prédire la probabilité d'une issue donnée.

Méthodes et algorithmes

Apprentissage supervisé On regroupe sous le terme de classifieurs linéaires tous les algorithmes qui cherchent à construire des hyperplans séparant les données pour correspondre aux classes fournies. On peut compter parmi eux différentes implémentations.

L'analyse factorielle discriminante [McLachlan, 2004] cherche à décrire, prédire et expliquer l'appartenance des données à une classe donnée. Elle regroupe divers algorithmes : l'analyse discriminante linéaire (LDA) [Izenman, 2008], l'analyse discriminante quadratique (QDA), l'analyse discriminante de Fisher (FDA) [Fisher, 1936] sont parmi les plus connues. Elles construisent des modèles génératifs.

Un classifieur naïf de Bayes [Rish, 2001] fait l'hypothèse de l'indépendance entre les variables et cherche à modéliser les probabilités conditionnelles d'appartenance à

une classe à partir de la règle de Bayes. Il s'agit également d'un modèle génératif.

Les réseaux de neurones utilisés en mode supervisé, comme par exemple le perceptron [Rosenblatt, 1958] avec un apprentissage par rétro-propagation [Rumelhart et al., 1986]. Ce modèle est discriminatif.

Les machines à vecteur de support ou séparateurs à vaste marge (SVM) [Boser et al., 1992] sont une généralisation des classifieurs linéaires qui cherche à construire des hyperplans à marge maximale séparant les données, et qui peuvent être utilisés à des fins de classification ou de régression, pas forcément sur des problèmes linéairement séparables grâce à une astuce consistant à ajouter des dimensions permettant la séparation linéaire. Il s'agit d'une méthode à modèle discriminatif.

Il existe d'autres classifieurs non linéaires, par exemple les classifieurs quadratiques, qui ne cherchent pas à construire des hyperplans mais des surfaces quadriques.

De même, les algorithmes de régression peuvent être linéaires ou non. Ici, la linéarité signifie que le modèle proposé sera une combinaison linéaire de divers facteurs. La différence entre les algorithmes linéaires réside principalement dans les différentes façons d'estimer les paramètres, comme par exemple en minimisant le carré de l'erreur, ou en cherchant à obtenir une vraisemblance maximale, ou encore par des méthodes bayésiennes. On utilise également souvent la régression logistique, qui utilise la fonction logistique pour modéliser la loi de probabilité.

Un arbre de décision est un arbre qui, à chaque nœud, indique, en fonction de la distribution d'une variable donnée, les résultats sur la distribution de probabilité d'une variable cible. Comme son nom l'indique, il peut aider à la prise de décision [Rokach, 2008]. Les résultats peuvent prendre la forme d'une classification ou d'une régression. Cette structure peut être créée par apprentissage supervisé à l'aide de plusieurs algorithmes, dont les plus connus sont ID3 [Quinlan, 1986], C4.5 [Quinlan, 1993] et CART [Breiman et al., 1984].

Les méthodes ensemblistes consistent en l'utilisation de plusieurs algorithmes plutôt qu'un seul, pour moyenner leur résultat et ainsi minimiser l'erreur commise. Pour cela, on utilise la technique du bagging, abréviation de bootstrap aggregating, qui consiste en la séparation de l'ensemble d'apprentissage en plusieurs sous-échantillons, et celle du boosting, qui consiste en la pondération des algorithmes pour donner plus de poids à ceux qui ont les meilleurs résultats. Adaboost [Freund and Schapire, 1995] est un algorithme de boosting qui s'appuie sur ce principe, avec un paramètre de mise à jour adaptatif permettant de donner plus d'importance aux valeurs difficiles à prédire. L'algorithme des forêts d'arbres décisionnels [Breiman, 2001] est une méthode ensembliste basée sur l'utilisation d'arbres de décision.

La méthode des k plus proches voisins [Fix and Hodges Jr, 1951] choisit simplement comme sortie la sortie moyenne des k entrées d'apprentissage les plus proches de l'entrée proposée. Il s'agit d'une forme de raisonnement par cas. Cette méthode peut être utilisée pour la classification ou pour la régression.

Apprentissage non supervisé Pour faire de la prédiction structurée, les modèles de Markov cachés permettent de modéliser une séquence en tant que chaîne de Markov (également un modèle génératif probabiliste), grâce à des algorithmes d'apprentissage comme l'algorithme de Baum-Welch [Welch, 2003], qui est une généralisation à ces modèles d'EM, ou l'algorithme de Viterbi [Viterbi, 1967].

L'inférence bayésienne est une méthode d'inférence permettant de déduire la probabilité d'un événement à partir de celles d'autres événements déjà évaluées, à partir d'observations et d'un modèle des probabilités conditionnelles de chacun des événements. Elle s'appuie principalement sur le théorème de Bayes. À partir de cela, les réseaux bayésiens dessinent un graphe entre les variables reliées par une loi de probabilité conditionnelle. Ces réseaux peuvent être construits déjà pré-structurés et avec apprentissage des lois de probabilités, ou même avec apprentissage de la structure. Il s'agit là aussi d'un modèle servant à la prédiction structurée.

Comme son nom l'indique, la modélisation causale cherche à établir les liens causaux entre les différentes variables d'un système. Elle va donc plus loin que les autres formes de modélisation, qui cherchent habituellement à mesurer une forme de corrélation. Une forme de modélisation causale couramment employée se base sur les réseaux bayésiens, en leur rajoutant la contrainte que les liens représentés doivent être uniquement des liens causaux.

Une autre technique est la modélisation d'équations structurelles (*Structural Equation Modeling* ou SEM en anglais), qui représente les hypothèses causales comme les variables d'une équation. Elle permet de construire des variables cachées là où elles se révèlent nécessaire à la modélisation. L'analyse factorielle, l'analyse de chemins et la régression sont des exemples particuliers de SEM. L'analyse de chemin cherche à décrire les dépendances entre plusieurs variables, dans un graphe acyclique orienté. Pour cela, Sewall Wright a fait remarquer [Wright, 1921] qu'entre trois variables, seules trois structures de causalité étaient possibles :

1. $X \rightarrow Y \rightarrow Z$
2. $X \leftarrow Y \rightarrow Z$
3. $X \rightarrow Y \leftarrow Z$

Les cas 1 et 2 présentent tous deux la propriété d'indépendance conditionnelle de X et de Z sachant Y, contrairement au 3e cas. Cela signifie qu'une étude de données transversale (les sujets sont examinés à un seul point dans le temps) peut suffire à distinguer le 3e cas, mais que pour séparer les deux premiers, seule une étude longitudinale peut faire l'affaire. Cela a donné lieu à un algorithme d'apprentissage [Rebane and Pearl, 2013] proposé notamment par Judea Pearl, qui est un des fondateurs de cette discipline [Pearl, 2000].

D'autres algorithmes et méthodes se rattachent au cadre de la SEM, comme par exemple le modèle causal de Neyman-Rubin [Rubin, 1974]. Le test de causalité de Granger permet de tester si une série de données permet d'en prédire une autre. [Granger, 1969]

Une autre branche de la modélisation causale est la modélisation causale dynamique. Contrairement aux réseaux bayésiens, le graphe proposé n'est pas néces-

sairement acyclique, et contrairement à la SEM, elle ne nécessite pas d'hypothèse d'absence de corrélation temporelle dans les fluctuations des données. Son but est en effet de modéliser des systèmes dynamiques, qui peuvent comprendre des boucles de rétroaction. Cette méthode est principalement utilisée pour la modélisation du cerveau. [Kiebel et al., 2008]

Certains modèles de réseaux de neurones ont été conçus spécifiquement dans le but d'un apprentissage non supervisé, comme les cartes auto-organisatrices [Kohonen, 1984] ou le gaz neuronal [Martinetz et al., 1991], qui fournissent une nouvelle représentation des données.

Différents algorithmes de partitionnement existent. Le partitionnement hiérarchique, partant du haut (divisif) ou du bas (agglomératif), sépare les données les plus éloignées ou regroupe les données les plus proches, créant ainsi un dendrogramme qui peut être coupé au niveau souhaité pour obtenir le nombre de groupes souhaité. L'algorithme des k-moyennes [Steinhaus, 1957] effectue une itération entre une partition de Voronoï et le calcul de la moyenne de chaque groupe. Les algorithmes DBSCAN [Ester et al., 1996] et OPTICS [Ankerst et al., 1999] se basent sur la densité des groupes.

L'algorithme EM (Espérance – Maximisation) [Dempster et al., 1977] cherche à construire un modèle de mélange gaussien (modèle génératif probabiliste) avec le maximum de vraisemblance, en faisant l'hypothèse qu'il existe des variables sous-jacentes, non observables, qu'il cherche donc à trouver et modéliser. Il comporte itérativement une étape où l'on calcule l'espérance de la vraisemblance à partir des paramètres actuels et une étape où l'on choisit les paramètres en maximisant la vraisemblance trouvée à l'étape précédente. De même, il permet de réaliser du partitionnement.

Pour faire de l'apprentissage d'associations, les algorithmes les plus utilisés sont Apriori [Agrawal et al., 1994] et Eclat [Zaki, 2000].

La détection d'anomalie peut se baser sur d'autres techniques, comme le partitionnement (un point qui ne correspond à aucun groupe) ou l'apprentissage d'associations (une donnée qui ne correspond à aucune règle d'association). Des algorithmes spécifiques existent également, comme Local Outlier Factor [Breunig et al., 2000], qui compare la densité autour d'un point à la densité autour de ses plus proches voisins.

La fouille de motifs séquentiels est un domaine de l'exploration de données qui recherche des motifs significatifs entre des données qui arrivent en séquence. À la différence des séries temporelles, on ne fait pas l'hypothèse que le temps entre deux données est constant. L'algorithme GSP [Srikant and Agrawal, 1996] (*Generalized Sequential Pattern* en anglais, soit motif séquentiel généralisé) est un des algorithmes permettant la fouille de motifs séquentiels. Il est basé sur l'algorithme Apriori, à la base conçu pour l'apprentissage de règles d'association, qui recherche les données les plus fréquentes dans la base de données, puis les élargit afin de constituer des ensembles d'objets. Ici, GPS cherche les données les plus fréquentes et s'en sert de base à la création de séquences fréquentes de longueur croissante.

Apprentissage par renforcement L'apprentissage par différence temporelle [Sutton, 1988] est utilisé pour prédire le futur à l'aide d'un modèle utilisant le passé. On utilise les prédictions du futur proche, avec une pondération favorisant les plus proches, qui sont plus probablement correctes, pour prédire le futur plus lointain. Le Q-learning [Watkins, 1989] fonctionne de façon similaire, mais sa sortie est une action qui mène à une évolution de l'environnement et qui reçoit une récompense. On apprend donc à sélectionner la meilleure action dans une situation donnée. SARSA [Rummery and Niranjan, 1994] est une variante du Q-learning inspirée de la méthode de différence temporelle.

Les algorithmes évolutionnaires [Eigen, 1973] s'inspirent de la nature et plus spécifiquement du phénomène de l'évolution pour mettre en place un renforcement visant à sélectionner les individus correspondant le mieux à une fonction dite de fitness. Cela revient à une optimisation utilisant une métaheuristique. Les systèmes de classeurs [Holland and Reitman, 1978] utilisent les algorithmes génétiques pour sélectionner les meilleures règles de comportement.

3.2 Proposition d'utilisation de l'apprentissage automatique et de l'exploration de données pour le suivi des élèves

Après cette revue des méthodes proposées dans le domaine de l'extraction de données et de l'apprentissage automatique, et utilisées dans le domaine de l'e-learning, nous pouvons maintenant mieux définir la façon dont nous proposons d'utiliser ces méthodes pour améliorer le suivi des étudiants.

En effet, les indicateurs que nous avons présentés dans la section 2.2 nous permettaient déjà de mieux analyser nos données pour comprendre le positionnement d'un étudiant par rapport aux autres, et les statistiques présentées nous permettaient de mesurer son activité en termes absolus, ce qui permettait déjà de mieux le suivre. Toutefois, nous avons vu toute la puissance des méthodes d'extraction de données et d'apprentissage automatique pour proposer divers types de sorties très intéressantes permettant d'examiner nos données plus en profondeur. Nous avons vu également que l'utilisation de telles méthodes à des fins proches des nôtres était déjà courante.

Par ailleurs, le processus de définition de ces indicateurs et statistiques correspondait justement à la phase de pré-traitement nécessaire à un tel traitement, en nous permettant d'éliminer les informations inutiles et garder des données correspondant aux aspects que l'on souhaite étudier. En effet, les indicateurs que nous avons définis pourront, le cas échéant, servir de caractéristiques d'entrée aux diverses méthodes que nous allons proposer. Nous avons mentionné que ces indicateurs ne sont pas indépendants entre eux. Toutefois, dans un premier temps, nous allons tenter de les utiliser tels quels, et si les résultats obtenus ne sont pas satisfaisants, nous pourrions passer par une analyse en composantes principales.

Nous allons donc lister ici des propositions visant à utiliser l'apprentissage automatique pour améliorer le suivi des élèves. Ces propositions seront découpées par grand

type de méthodes. Nous ne souhaitons pas spécifier à ce stade de choix d'algorithme particulier. Au contraire, pour chacun des buts cités, il nous paraît pertinent de comparer plusieurs algorithmes entre eux et voir lequel donne les meilleurs résultats sur notre type de problème.

Partitionnement Le partitionnement cherche à découvrir des groupes ou des structures jusqu'alors inconnus dans les données. Nous pourrions l'utiliser pour établir des groupes d'étudiants au profil ou au comportement similaire, que nous chercherons à interpréter en termes de comportement des étudiants au sein d'un même groupe. Ces groupes pourront alors nous servir de base au suivi, s'ils correspondent peu ou prou au taux d'activité des élèves, ou aux notes obtenues aux activités, qui peuvent servir de base pour repérer que l'élève est démotivé ou en difficulté.

Nous pourrions alors comparer les groupes obtenus dans différentes formations pour vérifier lesquels se retrouvent dans plusieurs d'entre elles, et lesquels sont caractéristiques d'une seule formation. Si nous avons des groupes qui se retrouvent dans plusieurs formations, nous chercherons également à savoir si la proportion d'élèves dans ces groupes est la même dans toutes les formations. Une fois un certain nombre de profils types établis, on pourra déterminer auquel se rattache le plus un nouvel élève. En effet, il est également important de repérer les différentes méthodes d'apprentissage mises en œuvre par les étudiants pour pouvoir comprendre leurs spécificités et pouvoir leur proposer une offre de parcours adaptée.

Nous pensons également qu'il est intéressant d'étudier le nombre de groupes qui donne le meilleur résultat dans chaque formation. Est-il le même partout ? Si non, dépend-il du nombre d'élèves inscrits à la formation ?

Toutefois, avant de pouvoir exploiter les résultats, nous devons passer par une phase d'analyse permettant de comprendre à quel aspect de comportement correspondent les groupes obtenus. Nous pouvons penser voir apparaître des groupes reflétant le comportement des élèves sur le forum, mais aussi des élèves plus ou moins actifs, et des élèves ayant de plus ou moins bonnes notes. Il sera intéressant de voir à quel point ces variables sont corrélées ou non au sein des groupes proposés.

Classification La classification, qui cherche à appliquer une structure connue à de nouvelles données, peut nous permettre de décider si un élève fera partie de certains groupes que nous construirons nous-mêmes. Notamment, nous proposons de l'utiliser pour distinguer trois groupes : le groupe des élèves qui réussiront leur année de ceux qui iront jusqu'au bout mais échouons, et de ceux qui abandonneront en cours de route.

Il s'agit là d'une approche classique. Nous avons vu dans l'état de l'art de nombreuses méthodes visant à faire à peu près cette même analyse. Toutefois, le découpage en trois groupes que nous proposons n'est pas classiquement utilisé et est donc innovant. Par ailleurs, bon nombre d'études choisissent un seul algorithme pour cette tâche, alors que nous souhaitons en comparer le plus grand nombre pour identifier celui qui donne les meilleurs résultats.

Une fois les résultats obtenus, nous chercherons à étudier le classifieur constitué, si l'algorithme le permet, pour voir quels critères ont été majoritairement utilisés, et donc nous permettre de comprendre quels critères sont les plus importants pour la prédiction du succès des étudiants. Nous comparerons ce modèle avec celui obtenu par régression.

Nous observerons également le meilleur taux de succès obtenu. S'il est élevé, on pourra juger que nos caractéristiques, à savoir les indicateurs, suffisent à prédire avec succès la performance des élèves, et constituent donc une bonne base pour leur suivi. S'il est trop bas, nous chercherons à rajouter des données supplémentaires pour améliorer le taux de succès de cette méthode.

Régression La régression, qui tente de modéliser les données avec le moins d'erreur possible, peut nous permettre de tenter de modéliser la note finale d'un élève en fonction de son action durant l'année. Ainsi, après avoir construit un modèle sur les promotions passées d'une formation donnée, nous pourrions l'appliquer aux élèves en cours de cette même formation. On peut, au choix, modéliser l'abandon des élèves par l'obtention d'une note de -1, ou n'appliquer la régression qu'aux élèves n'ayant pas abandonné la formation ; et on pourra bien évidemment comparer les résultats de ces deux méthodes.

Là encore, nous tenterons d'analyser le modèle obtenu afin de comprendre quels critères sont importants pour prédire la note finale de l'élève. Si nous repérons qu'historiquement, un type de comportement a conduit à un échec ou un abandon de la formation, nous pourrions construire un indicateur de ce type de comportement chez les étudiants en cours de formation, qui nous permettra, dès le repérage, d'agir vite et d'en informer l'apprenant.

Apprentissage d'association L'apprentissage d'association, qui recherche des liens entre différentes variables, peut nous aider à remarquer les corrélations entre nos indicateurs et statistiques. Avec un point de vue légèrement différent, nous pourrions obtenir des informations semblables à celles données par le partitionnement, où les élèves ayant une caractéristique de comportement donnée en ont également fréquemment une autre.

Détection d'anomalie La détection d'anomalie, qui vise à identifier des données inhabituelles ou des erreurs, peut nous permettre de repérer un changement de comportement négatif d'un élève et de lui demander si ce changement est justifié, avant qu'il n'ait un impact négatif trop marqué sur sa réussite.

Nous avons découvert dans l'état de l'art du domaine de l'exploration de données éducatives des idées intéressantes d'utilisation de méthodes d'apprentissage automatique pour le suivi des étudiants avec le partitionnement, la classification, la régression et l'apprentissage d'association. Après nous être documentés sur les domaines de l'exploration de données et de l'apprentissage automatique, nous avons découvert une autre méthode qui nous semble pouvoir donner des résultats intéressants : la détection d'anomalie. Nous proposons donc de l'utiliser également.

Nous montrerons comment nous avons implémenté ces propositions dans la partie 5, et nous décrirons les résultats des expérimentations que nous avons menées dans la partie 5.4.

Nous avons également deux autres propositions que nous développerons dans des parties suivantes.

La première d'entre elles est l'utilisation de l'algorithme d'apprentissage automatique HTM-CLA, inspiré du cerveau humain, qui était un des objectifs de notre projet. Nous n'avons pas cherché à lister HTM-CLA dans cette partie car il nous semble avoir beaucoup de capacités, lui permettant avec un peu d'adaptation de recouvrir bon nombre des catégories que nous venons d'établir. Le chapitre suivant sera donc consacré à une explication plus détaillée de notre choix de cet algorithme, de son fonctionnement et des méthodes proches. Nous développerons ensuite une proposition pour son utilisation dans la partie 4.3.

La seconde est la poursuite de l'automatisation du travail des tuteurs et responsables de formation par la création d'un tuteur virtuel intelligent, également basé sur HTM-CLA, qui sera détaillée dans la partie 4.3.5.

Chapitre 4

Utilisation de HTM pour le suivi des étudiants

Dans ce chapitre, nous allons tout d'abord expliquer notre choix de nous intéresser à l'algorithme HTM-CLA pour notre problématique du suivi des étudiants. Une fois cet intérêt établi, nous expliquerons plus en détail cet algorithme, avec de brefs rappels biologiques sur le fonctionnement du cerveau humain, une présentation des domaines voisins des réseaux de neurones et de l'apprentissage profond, et finalement une explication détaillée des spécificités et des capacités de HTM-CLA. Enfin, la fin du chapitre nous verra proposer plusieurs façons différentes d'adapter et d'utiliser HTM-CLA pour le suivi des étudiants.

4.1 Le choix d'un algorithme d'apprentissage performant inspiré du cerveau humain

Nous présentons ici notre argumentaire pour l'utilisation de HTM-CLA. Cette explication se déroulera en deux parties. La première partie tentera d'expliquer l'intérêt des algorithmes cherchant à reproduire le fonctionnement du cerveau humain. La deuxième partie tentera de montrer la bonne adaptation des hypothèses fondatrices de HTM-CLA à notre problème.

4.1.1 Cognitivism et connexionnisme

Les humains possèdent de remarquables capacités d'apprentissage. En quelques années à peine, ils apprennent notamment la locomotion et le langage, deux facultés qui sont encore à ce jour difficilement reproductibles par des programmes informatiques. Il est donc compréhensible que la recherche d'algorithmes d'apprentissage efficaces passe par des tentatives d'imitation du fonctionnement des humains.

Deux courants principaux, le cognitivisme et le connexionnisme, s'affrontent au sein de l'intelligence artificielle générique. Le cognitivisme considère qu'il est important d'essayer de reproduire les mécanismes de pensée humains – la cognition,

et relève donc d'une approche top-down, de décomposition, assez abstraite, tandis que le connexionnisme vise l'imitation du cerveau lui-même, et relève donc davantage d'une approche bottom-up, de construction, plus concrète.

Le cognitivisme La cognition est définie comme l'ensemble des processus mentaux : tout ce qui ne relève ni de la perception, ni de l'action. On peut lister comme en faisant partie l'acquisition, la rétention et la manipulation du savoir. Le champ d'études associé à la cognition se nomme les sciences cognitives ; elles sont au carrefour de la psychologie, des neurosciences, de l'anthropologie, de la linguistique et de la philosophie.

Le cognitivisme s'est construit dans les années 1950 par opposition au paradigme dominant de l'époque, le béhaviorisme, qui considérait que la meilleure façon d'observer et de comprendre les processus mentaux était d'observer sa résultante, le comportement. Il est évident que cette approche néglige d'importants aspects des processus mentaux : imaginons par exemple quelqu'un qui lit un texte. Extérieurement, il ne fait rien, mais intérieurement, il emmagasine de l'information. Même l'apprentissage du langage ne peut se faire par un simple réflexe conditionnel pavlovien ni un conditionnement opérant du type de celui formalisé par Skinner, comme l'a avancé Noam Chomsky [Chomsky, 1959] ; il nécessite un état mental capable de généraliser les connaissances et surtout de générer de nouvelles phrases.

Le cognitivisme considère donc que la pensée est avant tout un processus de traitement de l'information, et s'intéresse aux états mentaux, considérés comme discrets. Il cherche également à décomposer le fonctionnement de l'esprit en briques élémentaires, selon une approche réductionniste. Cette théorie a notamment été proposée par Hilary Putnam en 1963 [Putnam, 1963], et son élève Jerry Fodor a poursuivi son développement [Fodor, 1975].

Ce point de vue a également été inspiré par les travaux de deux mathématiciens :

- La théorie de l'information de Claude Shannon [Shannon, 1948, Shannon and Weaver, 1949], qui démontre que les circuits électriques peuvent être interprétés en tant qu'opérations booléennes.
- La célèbre machine formelle d'Alan Turing, qui démontre comment une manipulation élémentaire (5 opérations, choisies par un ensemble de règles correspondant à des états) de symboles binaires suffit à représenter un grand nombre d'opérations, peut-être même tous les algorithmes, suivant la thèse de Church. [Church, 1935, Turing, 1950].

Ces deux thèses apportent la preuve qu'une machine est capable de raisonnement logique. Les cognitivistes y voient immédiatement un rapprochement avec le fonctionnement du cerveau. Ils pensent qu'on peut appréhender le système cognitif de la même manière.

Plus spécifiquement, on nomme computationnalisme le courant du cognitivisme qui considère que l'esprit humain se comporte de façon analogue à un algorithme ou un logiciel. Son fonctionnement est supposé relativement indépendant de la biologie du cerveau humain, qui n'est que l'ordinateur sur lequel sont effectués les calculs. Selon

ce point de vue, l'esprit humain manipule essentiellement de façon algorithmique, en employant des règles logiques, des structures symboliques. Les algorithmes peuvent être très différents suivant la tâche à effectuer : vision, raisonnement, calcul. . .

Ainsi, ils choisissent volontairement d'occulter les structures et phénomènes biologiques, les voyant comme un simple support, étant ce qu'il est dans le cerveau par le hasard de l'évolution, mais non nécessaire au fonctionnement du tout. La vision cognitive est parfaitement résumée par Jerry Fodor dans [Fodor, 1975] :

- Les connaissances sont des structures symboliques.
- La cognition consiste en la manipulation de ces structures.

Or, un problème sous-jacent à la représentation symbolique concerne la connexion des symboles avec le monde réel. Les traitements opérants sur les symboles sont inutiles ou presque si ces mêmes symboles ne font pas directement référence à des objets ou des caractéristiques du monde réel, ce que l'on retrouve sous le nom de « Symbol Grounding Problem », qu'on peut traduire par « problème de fondement des symboles ». Selon les symbolistes, il suffit pour cela de pouvoir connecter les symboles à l'environnement, à l'aide de capteurs ou de périphériques d'acquisition par exemple, ce qui leur apparaît comme un problème trivial. Pour Stevan Harnad [Harnad, 1990], cela est loin d'être le cas et la connexion des symboles avec le monde est intimement liée à la définition de la cognition elle-même.

Pour cette raison, l'approche symbolique est également critiquée par Searle, pour qui un programme ne peut obtenir que le point de vue syntaxique et non sémantique, et donc ne peut comprendre ce qu'il manipule. Pour le démontrer, il présente une expérience de pensée devenue célèbre, intitulée « Chinese room » [Searle, 1980], ou chambre chinoise. Imaginons une pièce dans laquelle est placé un homme armé d'un livre et d'un stylo. Dans le livre se trouve une liste de règles faisant correspondre certains symboles chinois à certains autres. L'homme ne peut communiquer avec l'extérieur que par le biais d'une feuille de papier qu'on lui fait passer. Sur cette feuille, un texte en chinois, qui pose une question. Dès qu'il reçoit la feuille, l'homme se met au travail, ouvre son livre et note sur la feuille d'autres symboles en suivant les règles fournies par le livre, puis rend la feuille à l'extérieur une fois son travail achevé. Il se trouve qu'il vient d'écrire un texte chinois fournissant exactement la réponse à la question qui lui était posée. Peut-on pour autant dire que cet homme, ou même cette pièce, comprend le chinois ?

De façon plus générale, les chercheurs en intelligence artificielle se sont vite aperçus d'une sorte d'inversion de l'échelle de difficulté des problèmes : plus une tâche leur semblait avancée, et relevant plus spécifiquement de l'intelligence humaine (mathématiques, planification, raisonnement logique, jeu d'échecs. . .), et plus elle était facile à faire reproduire par un programme informatique.

À l'inverse, les tâches sensorimotrices (reconnaissance visuelle d'un objet, d'un visage, reconnaissance auditive du langage, d'une voix, déplacement, lancer de ballon. . .) ou encore le langage leur paraissaient très simples, puisque présents également ou sous forme analogue chez de nombreux animaux et très jeunes enfants, et puisqu'aucun humain non handicapé ne ressent de difficulté particulière à les effectuer. Cependant, ils se sont vite rendu compte que les reproduire de façon informatique

était bien plus difficile qu'anticipé. Selon la légende, l'illustre pionnier de l'intelligence artificielle, Marvin Minsky, avait confié à un élève en 1966 la mission de résoudre le problème de la vision par un ordinateur en tant que projet d'été. . . Presque cinquante ans plus tard, cette discipline connaît encore de nombreuses difficultés, malgré la découverte de nombreuses techniques et une augmentation selon la loi de Moore des puissances de calcul disponibles.

Cette découverte a pris le nom de paradoxe de Moravec [Moravec, 1988], du nom de celui qui l'a formalisée dans les années 1980, en reconnaissant son côté contre-intuitif pour les humains. Selon Marvin Minsky [Minsky, 1985], une possible explication est que ces tâches sont si bien maîtrisées par le cerveau qu'elles se déroulent de façon inconsciente, et c'est ce qui fait que leur difficulté ne nous apparaît pas. Selon Moravec, la théorie de l'évolution permet également d'expliquer ce paradoxe. Les tâches sensorimotrices, en tant que fonctionnalités biologiques anciennes, ont été perfectionnées par les mécanismes évolutionnaires durant des millions d'années, et sont donc hautement adaptées à la fois à notre environnement et à notre corps. Les facultés de raisonnement, apparues très récemment sur le plan biologique, ne se sont pas encore perfectionnées autant. On peut rajouter à cela l'hypothèse qu'une plus grande faculté de raisonnement abstrait n'apporterait peut-être pas à un humain suffisamment d'avantage à la sélection naturelle pour qu'une réelle pression de sélection s'instaure.

David Marr, chercheur spécialisé dans la vision par ordinateur, rejette également les approches symboliques, argumentant que l'aspect physique de la vision est essentiel et doit sous-tendre tout mécanisme symbolique plus avancé. On peut citer parmi les caractéristiques communes des approches symboliques :

- Recherche dans l'espace d'états potentiellement longue
- Représentation aisée
- Faible capacité d'apprentissage ([Sun, 1999])

Le connexionnisme Quant à lui, le connexionnisme envisage la cognition comme le résultat d'une interaction globale de parties élémentaires et identiques interconnectées au sein d'un réseau. Les tenants du connexionnisme utilisent souvent des réseaux de neurones. Un tel réseau doit être capable d'apprentissage.

Le connexionnisme s'inspire davantage de sciences telles que les neurosciences et la biologie. Les partisans de cette théorie notent sa plausibilité biologique, puisqu'elle s'inspire du cerveau :

- Le paradigme connexionniste s'appuie sur des structures neuronales plausibles, alors que les symboles n'ont pas de fondements biologiques à ce jour.
- Il se focalise sur l'apprentissage et la rétention d'expérience dans l'environnement.
- Enfin, le connexionnisme réfute l'idée que l'activité mentale consiste en la manipulation de symboles explicites. Il assure que ceux-ci sont inutiles car les représentations implicites formées lors de l'apprentissage suffisent.

Comme le font remarquer Goertzel et Pennachin [Pennachin and Goertzel, 2007] :

« Un façon presque certaine de créer une intelligence artificielle générique serait de recopier exactement le cerveau humain, jusqu'au niveau de chaque atome, dans une simulation numérique. Certes, cela requerrait des scanners cérébraux et un matériel informatique bien au-delà de ce qui est actuellement disponible. Mais si l'on regarde la courbe de l'amélioration des scanners cérébraux et du matériel informatique, on s'aperçoit que cette approche pourrait bien devenir pertinente aux alentours de 2030 à 2050. » Cependant, cette approche resterait extrêmement frustrante, car ce n'est pas parce qu'on est capable de recopier un système à l'identique qu'on comprend les principes qui sous-tendent son fonctionnement.

Malgré tout, l'approche connexionniste est pour certains également cognitiviste, en ce qu'elle accepte la capacité de l'être à organiser ses connaissances. Elle peut ainsi être considérée comme une approche hybride, qui se situe tout simplement à un autre niveau d'interprétation. En effet, on peut arguer que si les connaissances ne sont pas stockées sous forme de structures symboliques à proprement parler, elles existent tout de même à un niveau d'abstraction plus élevé et correspondent à l'expression de « configurations neuronales ». Au même titre, les règles logiques, incompatibles avec les structures neuronales, peuvent être considérées comme une formalisation des processus manipulant celles-ci.

Les détracteurs du connexionnisme font remarquer qu'en suivant de trop près le fonctionnement du cerveau, on risque de ne pas pouvoir profiter de certaines bonnes propriétés inaccessibles aux systèmes biologiques, ou de reproduire des parties qui ne relèvent que de l'implémentation choisie par la nature et non du nécessaire à l'intelligence.

Il est à noter qu'à l'heure actuelle, la majorité des grands projets d'intelligence artificielle générique relèvent du cognitivisme. On peut notamment citer parmi eux Cyc [Lenat, 1995] (basé sur la logique des prédicats et une énorme base de données de connaissance applicable au monde réel), NARS [Wang, 2000] (basé sur de la logique floue), Soar [Laird, 2012] (qui produit des règles à la façon des systèmes experts), et OpenCog [Hart et al., 2008] (un système intégrant différents composants très variés qui font appel à un réseau sémantique de termes).

Cependant, le connexionnisme prend sa revanche dans le domaine de l'apprentissage, où ses capacités sont bien plus importantes que celles des approches symboliques. Il peut même apprendre des structures symboliques telles qu'un arbre et est plus robuste, capable de généralisation.

Il est à noter que des architectures hybrides mélangeant ces deux approches ont déjà été proposées. Cependant, il est plus difficile de concevoir de tels systèmes, puisqu'en plus de concevoir chaque module, il faut gérer leur interaction ; l'architecture en est donc plus complexe, de façon aisément compréhensible. En outre, l'apprentissage est également plus difficile dans ces modèles.

Pour les raisons évoquées plus haut, nous avons choisi de nous intéresser à l'apport potentiel des méthodes connexionnistes dans le domaine de l'apprentissage. Nous allons détailler leurs fondements théoriques dans la partie 4.2, qui rappellera quelques

réalités biologiques sur le fonctionnement du cerveau humain, avant de nous intéresser plus en détail aux différents algorithmes d'apprentissage proposés en imitation du cerveau : les réseaux de neurones et l'apprentissage profond.

L'apprentissage humain a deux caractéristiques particulières et importantes : il est généralement non supervisé, puisqu'il a essentiellement lieu à la découverte du monde environnant, et seulement très minoritairement dans un cadre scolaire et formel ; et il est incrémental, au fur et à mesure des expériences vécues. Pour ces raisons, nous porterons un intérêt tout particulier aux algorithmes d'apprentissage correspondant à ces deux critères.

Toutefois, avant cela, nous souhaitons expliquer pourquoi un algorithme particulier, HTM-CLA, nous semble particulièrement bien adapté à notre problématique. Cette explication sera complétée dans la partie 4.2 par une comparaison de HTM-CLA avec les algorithmes issus des domaines voisins.

4.1.2 Adaptation des hypothèses fondatrices de HTM à notre problème

Nous allons maintenant examiner si les diverses hypothèses faites par HTM-CLA pour être plus performant s'appliquent à notre problème, afin de déterminer s'il est pertinent pour nous de l'utiliser.

4.1.2.1 Le modèle mémoire-prédiction

Le modèle mémoire-prédiction (memory-prediction framework en anglais) propose un lien entre les approches cognitiviste et connexioniste en montrant comment le néocortex est capable d'extraire des informations sémantiques et symboliques à partir du fonctionnement biologique des neurones en réseau.

Il a été décrit pour la première fois dans le livre *On Intelligence* [Hawkins and Blakeslee, 2004], écrit par Jeff Hawkins, le fondateur de la compagnie Palm Computing et inventeur du Palm Pilot, qui a aussi longuement étudié les neurosciences.

Selon cette théorie, le fonctionnement du néocortex, siège principal de l'intelligence, serait étonnamment simple et tiendrait notamment en un unique algorithme. Son rôle serait de mémoriser par un encodage hiérarchique les motifs et séquences de motifs présents dans notre environnement, pour ensuite se baser sur ces séquences temporelles pour prédire l'avenir à plus ou moins long terme.

Un célèbre aphorisme intitulé « No free lunch » [Wolpert and Macready, 1997], dans le domaine de l'optimisation, énonce que pour certaines classes de problèmes, aucun algorithme ne peut être meilleur partout que les autres. En poussant un peu l'analogie, aucun modèle de réseau de neurones ne devrait pouvoir se distinguer totalement des autres, sauf en intégrant de nouvelles hypothèses pertinentes quant aux données qu'il a à modéliser. Reste à savoir bien calibrer ladite hypothèse pour avoir assez de spécificité pour pouvoir gagner en performance, mais assez de généralité pour conserver cette performance sur toute la variété de problèmes qu'on souhaite traiter.

L'ajout de connaissances sur son environnement est une approche utilisée par exemple dans le domaine de la planification, avec les Hierarchical Task Network (HTN) [Erol et al., 1994]. Cependant, ils utilisent une description du domaine dans lequel se trouvera le problème tellement précise qu'il ne reste plus qu'à instancier les diverses variables. Il est donc hors de question d'imaginer utiliser un même réseau pour résoudre un Freecell et un Solitaire... sans parler encore d'applications bien plus diverses.

Dans le domaine de l'intelligence artificielle et de la cognition, on trouve aussi les tenants de cette approche alors dite écologique, dont James Gibson, qui a le premier développé cette théorie en opposition radicale au cognitivisme. Selon lui, la perception et l'action qui en découle ne sont que des mécanismes d'adaptation d'un organisme à son environnement.

Selon le point de vue de Gibson [Gibson, 1979b], les traitements cognitifs sur des représentations mentales, explication cognitiviste de l'intelligence, seraient beaucoup trop coûteux en temps et en énergie. La théorie écologique part en effet d'un principe d'économie selon lequel le système le plus simple sera toujours favorisé à résultats égaux. Elle ne prétend pas expliquer pourquoi certaines espèces ont un comportement plus complexe que d'autres (il semble difficile d'expliquer pourquoi un dauphin est bien plus intelligent que le plancton, alors que tous deux partagent un même écosystème), mais soutient que chacune est parfaitement adaptée à son environnement.

Ce point de vue a été élaboré plus tard par William Clancey avec sa théorie de la cognition située [Clancey, 1997], qui étend à la cognition (y compris humaine) la théorie écologiste. En conséquence, toute manifestation de complexité dans l'intelligence ne peut in fine venir que de la complexité de l'environnement. On parle également d'énaction au sujet de cette idée.

On retrouve cette approche écologique dans le modèle mémoire-prédiction, qui repose entièrement sur l'hypothèse que l'environnement dans lequel tout humain évolue est fortement structuré et hiérarchisé, avec un certain nombre de régularités aussi bien spatiales que temporelles.

Parmi les régularités purement spatiales, on peut citer le fait qu'il fait plus chaud vers l'équateur qu'aux pôles. L'alternance du jour et de la nuit est une régularité uniquement temporelle pour qui ne voyage pas. Enfin, certains phénomènes ont des régularités spatio-temporelles, comme la météorologie d'un endroit donné : plus on s'éloigne dans le temps et l'espace, plus elle a des chances d'être différente. Comme point commun à tous ces phénomènes, on peut citer l'existence de causes sous-jacentes, qui ont pour cause ultime les lois de la physique, et qui permettent d'expliquer leur structuration.

Selon l'hypothèse de Hawkins, l'approche écologique est en fait suffisante pour expliquer le fonctionnement du cerveau humain. Le cerveau aurait donc pour fonction d'acquiescer un modèle mental de cet environnement complexe afin d'y adapter le comportement. Cette première définition peut sembler classique, mais la suite l'est moins : la structure cérébrale serait donc également parfaitement adaptée à ce rôle de modélisation et à l'encodage d'invariants spatio-temporel. C'est ce qui rend sa théorie à la fois simple, élégante et remarquablement puissante. Le reste de sa théorie

visé à expliquer les trois points suivants :

- quelle est la structure cérébrale permettant cette modélisation des invariants ;
- comment se fait leur apprentissage ;
- et comment ils sont ensuite exploités pour créer l'intelligence.

Ces trois points seront détaillés dans la partie 4.2.4, une fois que nous auront examiné les conséquences des hypothèses de la régularité spatiale et temporelle sur l'adaptation de HTM à notre problème.

4.1.2.2 Analyse de la contrainte de régularité spatiale et temporelle

Nous avons vu que HTM s'appuyait sur une contrainte supplémentaire par rapport à bon nombre de modèles similaires, correspondant à l'hypothèse d'une forte structuration spatiale et temporelle de l'environnement. Nous allons donc examiner l'adéquation entre ces contraintes et les données sur lesquelles nous sommes amenés à travailler.

Dans un livret intitulé « Problems that fit HTM » [Numenta, 2007], soit « Les problèmes qui conviennent aux HTM », Numenta prend le temps de détailler ces contraintes. Puisque les HTM cherchent à créer une hiérarchie spatio-temporelle, cette hiérarchie ne pourra correspondre à un modèle correct du monde que si ce monde est également régi par des causes hiérarchiques se suivant dans le temps.

Un exemple de problème qui correspond serait des capteurs météorologiques répartis sur une zone suffisamment large. Ces capteurs mesureraient tous des valeurs différentes, mais les capteurs plus proches mesureraient des valeurs plus proches. Des variations temporelles affecteraient les valeurs relevées par ces capteurs, et ce, à différentes échelles de temps, allant de la tempête jusqu'au changement de saison. Enfin, les changements à haut niveau, comme ceux des saisons, correspondent à la fois aux longues échelles de temps, mais également à une influence sur de grandes échelles spatiales. C'est l'inverse pour les causes de bas niveau, qui sont à la fois plus courtes et plus locales.

Ils proposent donc les critères suivants :

1. y a-t-il, à un instant donné, une forte corrélation entre les données générées par une même cause ?
2. les causes de haut niveau varient-elles plus lentement que les causes de bas niveau ?
3. peut-on imaginer une hiérarchie entre au moins quelques causes de notre problème ?
4. les données recueillies suffisent-elles à pouvoir déduire les causes ?
5. ce problème peut-il être résolu par l'application de règles simples, ou est-il plutôt flou et ambigu ?

Nous proposons les réponses suivantes à ces questions pour notre problème de suivi des élèves.

1. Question : Y a-t-il, à un instant donné, une forte corrélation entre les données générées par une même cause ?

Réponse : À un instant donné, les données que nous recueillons viennent des divers étudiants connectés. Nous allons donc avoir une première corrélation consistant en l'enregistrement de logs uniquement pour ces étudiants et non pour les autres. Ensuite, chaque étudiant, durant sa session, fait des activités en lien avec son but pour cette session. Il peut s'agir de lire une leçon en naviguant entre les différentes pages de cette leçon, ou de lire une leçon puis faire les activités associées. . . On voit que les ressources consultées et activités faites ne sont pas choisies au hasard par l'étudiant, ce qui nous donne une deuxième corrélation au sein des logs d'un étudiant donné. Enfin, on peut également espérer remarquer une certaine corrélation entre l'activité d'un étudiant et ses notes aux activités notées, ou entre ses échecs aux activités et ses actions suivantes (se renseigner sur un forum, retourner lire la leçon. . .), ce qui est donc une troisième corrélation.

2. Question : Les causes de haut niveau varient-elles plus lentement que les causes de bas niveau ?

Réponse : À haut niveau, nous avons la période de la formation, qu'on peut découper en début, milieu et fin. Au début, on peut supposer que tous les élèves travailleront, et que les situations d'abandon n'apparaîtront que progressivement. À la fin de la formation, on peut également penser retrouver un surcroît d'activité dû à l'approche des examens. À plus bas niveau, nous avons l'alternance des semaines, qui nous permet de supposer qu'un élève travaillera davantage certains jours que d'autres (en semaine s'il étudie à temps plein, et au contraire, le week-end sinon). À encore plus bas niveau, l'alternance des jours nous permet de faire l'hypothèse qu'un élève travaille plutôt le jour s'il étudie à temps plein, en soirée sinon. Parallèlement à cela, nous avons les différents cours proposés, dans lesquels l'étudiant va naviguer à son rythme. Dans un cours donné, il va probablement lire les leçons dans l'ordre chronologique, et ne pas faire les exercices avant d'avoir lu les leçons. Il aura peut-être plusieurs cours commencés en même temps, ou choisira au contraire de les faire en séquence. Nous avons donc bien une variété de causes qui peuvent expliquer des temporalités à différentes échelles dans notre problème, et où les causes à plus haut niveau correspondent aux variations temporelles les plus lentes.

3. Question : Peut-on imaginer une hiérarchie entre au moins quelques causes de notre problème ?

Réponse : Il suffit de quelques exemples pour montrer que les causes des logs qui s'enregistrent sont en hiérarchie. Par exemple, un étudiant peut avoir une mauvaise note à une activité par malchance sur les questions ou parce qu'il a mal compris la leçon (il aura donc des mauvaises notes également aux autres activités concernant cette leçon). Mais il peut avoir mal compris la leçon parce qu'il n'a pas assez travaillé cette leçon (les mauvaises notes y seront donc circonscrites) ou parce qu'il n'a pas assez travaillé en règle générale (il aura donc également des mauvaises notes aux activités correspondant aux autres leçons). Il peut n'avoir pas assez travaillé par manque de motivation (cette

situation risque de perdurer) ou par manque de temps (la situation pourra s'améliorer ou non) ou parce qu'il a été malade (la situation s'améliorera certainement).

4. Question : Les données recueillies suffisent-elles à pouvoir déduire les causes ?

Réponse : Dans les exemples donnés en réponse à la question précédente, nous estimons que les notes corrélées aux activités et étagées dans le temps peuvent aider à identifier les différentes situations que nous avons listées. Toutefois, nous apporterons ici deux bémols : le premier est qu'il peut falloir un certain temps afin que l'évolution ou non de la situation permette d'identifier les causes sous-jacentes. Le second est que l'apport de données supplémentaires aiderait à clarifier la situation (par exemple l'entretien d'un tuteur avec son élève).

5. Question : Ce problème peut-il être résolu par l'application de règles simples, ou est-il plutôt flou et ambigu ?

Réponse : Nous avons expliqué précédemment que jusqu'ici, les responsables de formation ne pouvaient suivre les élèves qu'à partir de règles simples, mais que ces règles péchaient par leur excès de simplicité. Non seulement il nous faudrait rajouter une grande quantité de règles pour correspondre à tous les cas possibles d'activité d'un élève parmi les différents volets, mais en outre, il n'existe effectivement pas de seuil bien défini permettant de déterminer si un élève est en situation d'échec ou d'abandon. C'est ce qui fait toute la difficulté et l'intérêt du problème.

Étant donné l'ensemble de nos réponses à ces questions, nous pouvons donc valider notre intérêt pour les HTM comme méthode d'apprentissage utilisable dans le cas de notre problème. Sur les cinq questions proposées, nous avons obtenu des réponses très positives à quatre d'entre elles, et une réponse plutôt positive à la dernière. Les deux bémols que nous avons apportés en réponse à la question 4 nous semblent être mitigés : le premier nous semble en effet inhérent à l'utilisation des HTM, et le deuxième pourra éventuellement être corrigé, s'il s'avère problématique, par le recueil et l'intégration de nouvelles données.

Ces hypothèses nous semblent donc bien correspondre aux données et phénomènes que nous cherchons à modéliser, expliquer et prévoir dans le cadre de notre problématique. Pour toutes ces raisons, nous avons donc retenu HTM-CLA comme le modèle d'apprentissage potentiellement le plus pertinent pour notre problème.

Nous allons maintenant détailler les algorithmes voisins de HTM afin de bien comprendre ce qui le distingue. Suite à cela, nous pourrions étudier l'algorithme HTM lui-même, plus spécifiquement dans sa version Cortical Learning Algorithm.

4.2 L'apprentissage inspiré du cerveau humain

4.2.1 Fonctionnement du cerveau humain

Dans cette partie, nous allons faire de brefs rappels sur le fonctionnement du cerveau humain. On pourra trouver une version plus détaillée de cette partie en

annexe C, dont les détails poussés en termes de biologie nous ont semblés intéressants pour mieux comprendre l'équivalence biologique de certains concepts de HTM-CLA. Toutefois, cette compréhension n'étant pas obligatoire à celle du fonctionnement de l'algorithme global, sa lecture pourra être réservée aux plus intéressés.

Le cerveau humain Le cerveau est le siège de l'intelligence humaine. Il reçoit et traite les perceptions venant des nerfs et renvoie des ordres d'actions par ce même canal. Le cerveau se décompose en plusieurs parties, dont la dernière apparue à l'échelle de l'évolution est le néocortex, probable siège de l'intelligence spécifique aux humains. Le cerveau est essentiellement composé de neurones reliés entre eux par des synapses.

Le neurone Les neurones reçoivent et transmettent un signal électrique, appelé influx nerveux ou potentiel d'action. Ces signaux arrivent par les synapses d'entrée du neurone, appelées dendrites, et repartent par les synapses de sortie, situées au bout d'un long filament appelé axone. Ils prennent la forme d'impulsions séparées par des périodes sans signal. Pour que le signal se transmette à travers le neurone, il faut que la somme des entrées dépasse un certain seuil. Une fois que le neurone a transmis une impulsion, il ne peut plus le faire pendant une certaine période dite réfractaire.

La synapse Les synapses sont les jonctions entre les neurones. Dans leur grande majorité, les synapses ne transmettent pas le signal nerveux sous forme électrique mais chimique, avec libération d'ions par le neurone présynaptique lors de l'arrivée de l'influx nerveux, qui sont captés par le neurone postsynaptique et lui provoquent une dépolarisation impulsionnelle.

L'apprentissage Le cerveau apprend en se modifiant à plusieurs niveaux. La plasticité synaptique est la faculté des synapses à changer la force de leur effet sur les neurones. La plasticité neuronale est la faculté des neurones à modifier leur excitabilité. Enfin, la plasticité cérébrale désigne les modifications des réseaux de neurones par l'apparition ou la disparition de synapses entre eux.

La loi de Hebb, bien connue, propose un modèle du phénomène de plasticité synaptique [Hebb, 1949]. Selon cette règle, quand une cellule A excite par son axone une cellule B et que, de manière répétée et persistante, elle participe à la genèse d'une impulsion dans B, un processus de croissance ou un changement métabolique a lieu dans l'une ou dans les deux cellules, de telle sorte que l'efficacité de A à déclencher une impulsion dans B est, parmi les autres cellules qui ont cet effet, accrue. On la résume souvent par la phrase "Cells that fire together, wire together".

Structure du néocortex Les neurones sont disposés dans le néocortex en six couches et de très nombreuses colonnes. Selon la théorie de Mountcastle, la colonne

est l'unité fonctionnelle basique du cerveau [Mountcastle, 1997], l'élément de base d'un algorithme massivement parallèle.

Codage de l'information On sait depuis bien longtemps que les neurones, lorsqu'ils sont activés, déclenchent une impulsion électrique qui va être communiquée aux neurones qui leur sont reliés. Cependant, la façon dont ces impulsions sont reliées aux stimuli qui les ont déclenchées, c'est-à-dire la façon dont l'information est représentée dans le cerveau est, elle, encore bien floue. On sait que les impulsions électriques peuvent varier en durée, en amplitude, en profil, mais surtout en durée qui les sépare.

Non seulement il peut être extrêmement intéressant de mieux savoir comment le cerveau représente et traite les informations afin d'en déduire des limitations ou avantages à notre perception du monde, mais si nous connaissions parfaitement la méthode d'encodage utilisée, on peut imaginer ensuite qu'un décodage serait possible. On pourrait alors, en visualisant le flux d'activité neuronale dans le cerveau, savoir ce que perçoit, ressent, pense ou imagine la personne analysée. Ces applications peuvent sembler futuristes, néanmoins elles en sont déjà à leurs balbutiements : ainsi, une équipe de l'université de Berkeley a pu reconnaître quelques mots entendus par leurs sujets de test et même les reconstruire [Pasley et al., 2012], et reconstruire également des vidéos regardées par d'autres sujets [Gallant, 2015]. Parmi les possibles utilisations concrètes de ces technologies, on peut penser à la commande de divers objets simplement par la pensée, ou encore à l'aide aux personnes entièrement paralysées mais conscientes.

Parmi les hypothèses en concurrence actuellement, les deux principales sont le codage en taux et le codage temporel. Le codage en taux correspond à l'hypothèse que l'information transmise est le nombre d'impulsions par unité de temps, tandis que le codage temporel s'intéresse au moment exact où chaque impulsion est déclenchée. Les neurones moteurs semblent piloter les muscles par un codage en taux [Adrian and Zotterman, 1926]. Toutefois, en plus de négliger l'information du moment de déclenchement des impulsions, ce codage peut être trop simpliste pour correspondre au fonctionnement du cerveau [Stein et al., 2005]. D'autre part, la très grande rapidité de réponse à certains stimuli (par exemple la reconnaissance visuelle) semble indiquer que les neurones n'ont pas le temps, avant de répondre, de calculer des moyennes temporelles ; et on a vu que le moment précis de déclenchement d'une impulsion pouvait avoir un grand impact, par exemple en tombant ou non dans la zone d'insensibilité post-excitatoire du neurone suivant, ou dans une étude montrant qu'une seule impulsion pouvait avoir un véritable effet papillon [London et al., 2010]. En outre, le codage temporel associé à la méthode de *spike timing dependent plasticity* a donné de très bons résultats de simulation [Thorpe, 1990].

Une autre dimension du codage neuronal est le pourcentage de neurones activés à un instant donné par rapport au nombre total de neurones. Si un stimulus est représenté par un seul neurone, on parle de codage local. S'il est représenté par un faible pourcentage des neurones disponibles, on parle de codage clairsemé (ou parcimonieux, suivant les traductions de *sparse coding*). Enfin, si environ une moitié

des neurones est activée, on parle de codage dense. Notons qu'il ne sert à rien de considérer un pourcentage de neurones supérieur à $1/2$, dans la mesure où on peut retomber dans les catégories définies plus haut en inversant les neurones activés et non activés, ce qui ne fait pas perdre d'information. Parmi ceux-ci, le codage clairsemé est le plus probablement utilisé dans le cerveau. Il présente des propriétés intéressantes qui sont un bon compromis entre les deux autres types d'encodage.

En effet, un codage local ne permet de représenter que N données avec N neurones, ce qui n'est pas réaliste si l'on compare le nombre de données stockées dans le cerveau au nombre de neurones qui y figurent. En outre, ne présentant aucune redondance, ce codage n'est pas du tout robuste. En revanche, il est très facile d'associer à un seul neurone un ensemble d'entrées, et donc d'apprendre, d'encoder et de décoder. Il présente également l'avantage qu'il est facile de reconnaître plusieurs motifs activés simultanément, et qu'aucune interférence ne viendra perturber la reconnaissance. Toutefois, la généralisation n'est pas possible, car ce codage n'est aucunement modulaire.

Un codage dense, ou holographique, lui, permet de représenter 2^N données avec N neurones, et puisqu'il en emploie un grand nombre à la fois, présente une grande robustesse et des capacités de généralisation. Ici toutefois, une telle capacité de mémoire est complètement extravagante au regard des besoins du cerveau, dont on ne voit donc pas pourquoi il aurait eu besoin de créer tant de neurones inutiles. En revanche, il est extrêmement difficile et lent d'apprendre avec un tel codage, puisque les liens faisant correspondre entrées et sorties doivent nécessairement être très complexes, et probablement non linéairement séparables. On peut donc dire adieu aux règles simples et locales, comme la règle de Hebb. En outre, il est absolument impossible de superposer deux motifs sans introduire d'ambiguïté, puisque toute la population de neurones participe au codage.

Un codage clairsemé représente un bon compromis entre le codage local et le codage dense. Il a donc une bonne capacité de mémoire (bien que moindre que celle du codage dense, elle est amplement suffisante), une bonne robustesse à l'erreur, des capacités de généralisation, la possibilité de représenter plusieurs données à la fois, ainsi qu'une facilité d'apprentissage.

Une autre indication pointant vers un codage en tout cas distribué nous est fournie lors de l'examen des neurones de l'aire IT. Cette aire est le premier lieu de la reconnaissance visuelle des objets, toutefois, les neurones qui la composent ne reconnaissent pas eux-mêmes ces objets, mais plutôt un certain nombre de leurs caractéristiques. Ainsi, les neurones d'IT qui sont le plus sélectifs sur une forme le sont le moins sur les autres caractéristiques, comme la position, l'échelle, le contraste, le bruit... [Zoccolan et al., 2007] Cela semble signifier que la reconnaissance visuelle des objets passe par une liste de ces caractéristiques, correspondant donc à une représentation distribuée. Les neurones grand-mère seraient une population spécifique correspondant à un plus haut niveau d'abstraction où ils représenteraient également des éléments divers des perceptions, ce qui n'implique donc pas une représentation différente en nature aux différents niveaux d'abstraction. La représentation clairsemée est également tout indiquée, toujours dans le cortex visuel, par les expériences de

[Desimone et al., 1984, Kiani et al., 2007, Rolls and Tovee, 1995].

Plus généralement, même ailleurs dans le néocortex, les chercheurs ont généralement du mal à trouver, pour un neurone donné, quel stimulus peut l'activer [R Baddeley, 1997]. Les neurones ont donc une sélectivité trop importante pour relever d'un codage dense. Les distributions d'activité relevées dans le néocortex semblent correspondre, du point de vue de la théorie de l'information, à un codage clairsemé [Olshausen and Field, 2004, Földiák and Young, 1995].

Le langage nous fournit également un outil important d'introspection sur notre représentation des concepts. Comme chaque phrase fait appel uniquement à une faible fraction de tous les mots et concepts existants, il s'agit là d'un indice de plus pointant vers une représentation mentale clairsemée.

Dans cette partie, nous avons présenté les structures anatomiques du cerveau, en insistant principalement sur le néocortex, qui nous paraît la zone la plus représentative des spécificités humaines. Nous avons également indiqué comment ces structures participent aux fonctions du cerveau qui nous intéressent, à savoir la mémoire et l'apprentissage. Nous allons voir dans les parties suivantes les modèles informatiques d'apprentissage automatique qui ont été inspirés par ce fonctionnement.

4.2.2 Réseaux de neurones

De la même façon que pour les algorithmes génétiques, des informaticiens se sont inspirés de la biologie pour créer des réseaux de neurones formels afin de réaliser certaines opérations. Un neurone formel est une entité dont le comportement peut généralement être représenté par une fonction mathématique et qui est relié à d'autres neurones formels en un réseau.

Il est à noter que malgré l'inspiration biologique des neurones formels et des réseaux de neurones, leur comportement n'est généralement pas entièrement équivalent à celui des neurones biologiques ni du cerveau humain. Cela ne signifie pas que ces réseaux ne soient pas utiles pour autant : ils ont pu obtenir de bons résultats d'apprentissage.

Nous allons présenter ici une description rapide du domaine des réseaux de neurones. Le lecteur intéressé pourra trouver davantage de détails en annexe D.

Modèles de neurone formel Le neurone formel possède généralement plusieurs entrées et une sortie qui correspondent respectivement aux dendrites et à l'axone du neurone. Il est coutumier d'attribuer des poids aux entrées afin de représenter l'importance des synapses et de pouvoir effectuer des opérations plus complexes. Ces poids correspondent au nombre de récepteurs post-synaptiques. Le neurone formel réalise ensuite le calcul de la valeur de la sortie à partir des valeurs des entrées.

Le premier modèle de neurone formel a été celui de McCulloch et Pitts [McCulloch and Pitts, 1943], proposé dès 1943. Dans leur modèle, les entrées comme les sorties sont binaires (égales à 0 ou 1). Le neurone réalise une somme pondérée des entrées. La fonction de Heaviside est ensuite utilisée afin de vérifier si cette somme pondérée est ou non au dessus d'un certain seuil. Un schéma de ce modèle est présenté dans la figure 4.1.

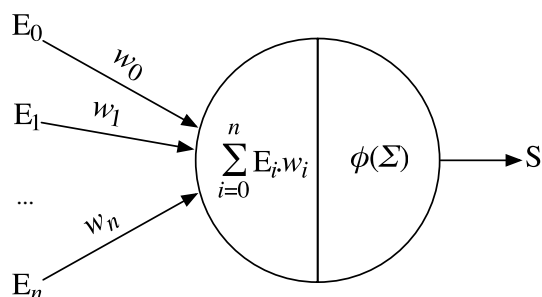


FIGURE 4.1 – Schéma d'un neurone formel

D'autres modèles plus réalistes ont pu être présentés par la suite. Nous mentionnons quelques-uns d'entre eux en annexe D.

Topologies des réseaux En 1957, Frank Rosenblatt crée le premier réseau de neurones à partir du neurone de McCulloch et Pitts : le Perceptron [Rosenblatt, 1958]. Il s'agit d'un réseau très simple, où les neurones sont disposés en seulement trois couches (une couche d'entrée, une couche de sortie, et une couche intermédiaire), dans lequel les connexions se font uniquement vers l'avant et dont la topographie ne peut pas évoluer. En revanche, les poids synaptiques entre la couche intermédiaire et la couche de sortie peuvent être modifiés afin de parvenir au résultat souhaité. La seule temporalité est introduite au moment du franchissement d'une synapse, qui induit un retard. Ce réseau est capable de réaliser les mêmes opérations qu'un classifieur linéaire, et a également été montré équivalent à un automate à états finis. Un exemple de Perceptron est présenté à la figure 4.2.

Les réseaux multicouche, rajoutent des couches de neurones (parfois dites « cachées ») entre les entrées et les sorties. On montre un exemple de topologie à deux couches cachées à la figure 4.3.

Dans tous les réseaux de neurones que nous avons présenté précédemment, les liens entre neurones n'allaient que dans un seul sens, de l'entrée vers la sortie, et n'allaient que d'une couche à la suivante. Il existe pourtant un autre type de réseau, développé dans les années 1980, qui comprend des liens dans les deux sens. On les appelle les réseaux récurrents.

Un réseau de neurones est dit récurrent s'il existe au moins un cycle dans la structure des liens entre neurones. Le fonctionnement d'un neurone en lui-même ne diffère pas nécessairement de celui des neurones des modèles antérieurs. L'existence de liens à l'envers permet de créer dans le réseau un comportement dynamique, mais lui rajoute aussi une fonction de mémorisation. Ainsi, si l'on voulait créer un réseau de neurones capable de réaliser une addition, seule une connexion vers l'arrière pourrait permettre de mémoriser la retenue et l'utiliser à l'étape suivante de l'addition. De tels réseaux sont donc plus complexes, mais également beaucoup plus

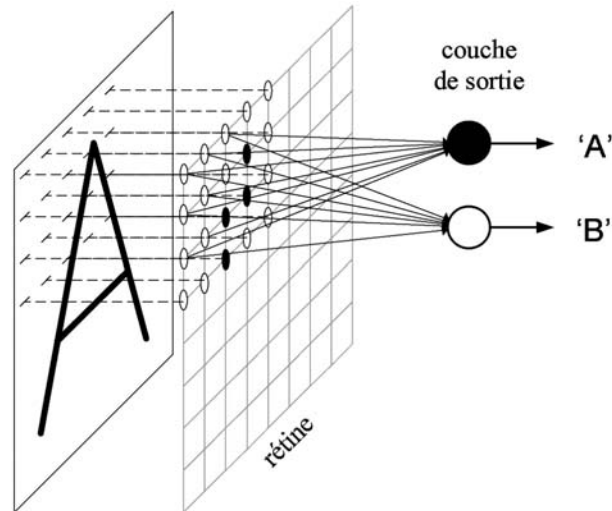


FIGURE 4.2 – Le Perceptron employé pour la vision

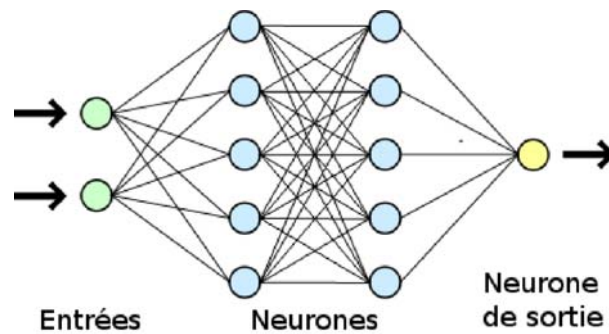


FIGURE 4.3 – Un réseau de neurones multicouche

puissants. Un exemple de réseau de neurones récurrent est présenté dans la figure 4.4.

On qualifie alors de *feedforward* ou directes les liaisons allant dans le sens de l'entrée vers la sortie, et de *feedback* ou rétroaction les liaisons allant dans le sens inverse.

Quelques topologies plus complexes sont détaillées en annexe D.

Les réseaux de neurones peuvent en outre être synchrones (tous les neurones déclenchent leurs impulsions ou non simultanément à chaque pas de temps) ou asynchrones (chaque neurone fonctionne à son rythme). Dans le premier cas, la modélisation est nettement simplifiée par rapport au deuxième, au détriment de la précision.

Méthodes d'apprentissage Les premières méthodes d'apprentissage des poids synaptiques utilisées par le Perceptron sont inspirées de la règle de Hebb, mentionnée

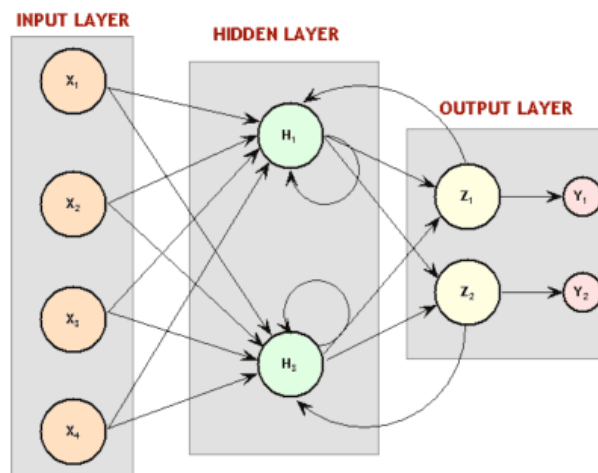


FIGURE 4.4 – Un exemple de réseau de neurones récurrent

plus haut, qui explique comment le déclenchement d'un neurone par un autre peut renforcer les liens qui ont mené à cet effet. Un théorème a prouvé qu'un réseau de neurones apprenant par la règle de Hebb convergerait nécessairement.

L'Adaline [Widrow and Hoff, 1960], d'une architecture assez similaire au Perceptron, réalisait son apprentissage par une descente en gradient visant à minimiser l'erreur quadratique entre la sortie réelle et la sortie théorique. Elle utilisait la fonction identité et non la fonction de Heaviside comme fonction d'activation d'un neurone.

Un autre algorithme très employé est celui de la rétropropagation du gradient de l'erreur [Bryson Jr, 1969, Rumelhart et al., 1986], qui est aussi une forme de descente en gradient, applicable également aux réseaux multi-couches. On applique le gradient de la fonction d'activation à la différence entre la sortie voulue et la sortie réelle. On fait alors remonter cette valeur à travers le réseau à l'envers, associant à chaque neurone une différence, dont on se servira ensuite pour modifier les poids de chaque connexion.

L'inconvénient de cette méthode est sa complexité en $O(c^3)$, où c est le nombre de connexions entre neurones du réseau. D'autre part, il n'existe pas de règle permettant de savoir à l'avance comment dimensionner un réseau par rapport au problème que l'on souhaite traiter. La seule règle qui existe est complètement empirique et ne permet que de calculer le nombre optimal de neurones cachés par rapport aux autres : $nb_exemples * erreur / (nb_entrees + nb_sorties)$

Rajouter des unités cachées peut permettre de réduire le nombre de minima locaux de la fonction d'erreur, dans lesquels on ne désire pas tomber, mais à partir d'un certain seuil, cela risque au contraire d'en augmenter le nombre par surapprentissage. Enfin, il faut préciser qu'il est nécessaire de normaliser et décorrélérer les données que l'on souhaite donner en entrée au réseau, sous peine d'avoir des difficultés voire des impossibilités d'obtenir une convergence de l'apprentissage.

Dans les modèles de réseaux de neurones que nous avons présenté jusqu'alors,

l'apprentissage était généralement supervisé. Cela signifie que le réseau était entraîné sur un certain nombre de paires entrée-sortie connues, jusqu'à arriver à les restituer correctement ; si les exemples d'apprentissage ont été bien choisis, on peut supposer que la généralisation se fera correctement et que la sortie proposée par le réseau de neurones pour une nouvelle entrée sera la bonne.

Une autre méthode, l'apprentissage non supervisé, s'offre en alternative. Elle consiste à présenter au réseau un certain nombre d'exemples représentatifs mais sans les associer à une contrainte sur la sortie. Le réseau répartira donc de lui-même les entrées les plus similaires sur les mêmes sorties. C'est cette méthode qui est généralement employée dans les réseaux de neurones auto-organisés.

Afin de mettre en œuvre cette méthode, le mécanisme généralement employé est de mettre les neurones en compétition entre eux. Seul le neurone le plus similaire aux données d'entrée sera réellement activé. Un neurone finira donc par se spécialiser dans la reconnaissance de caractéristiques précises. En effet, lorsque le neurone gagne la compétition, les poids qui le relient aux entrées sont modifiés de façon à être redistribués vers les entrées actives à cet instant.

Un exemple de réseau auto-organisé est la carte auto-adaptative, créée par Teuvo Kohonen en 1984 [Kohonen, 1984], dont le but est de cartographier un espace réel à des fins de classification. Son fonctionnement revient à découper l'espace des données d'entrée en différentes zones pertinentes qui respectent la topologie de cet espace. Au niveau de l'implémentation, les neurones sont placés dans une grille à deux dimensions (Kohonen recommande l'utilisation d'une grille hexagonale). À chaque neurone est associé son vecteur référent de poids auquel sera comparée toute nouvelle entrée ; le neurone dont le vecteur référent sera jugé le plus proche sera alors le seul activé (winner takes all). Afin de respecter la topologie, l'apprentissage n'est pas fait uniquement sur le neurone gagnant mais également, à moindre dose, sur ses voisins. Cette méthode a été comparée à une généralisation non linéaire de l'Analyse en composantes principales (ACP). La figure 4.5 présente un exemple d'utilisation de carte auto-adaptative pour classer des couleurs.

Résultats théoriques Il a été prouvé que le neurone formel de McCulloch et Pitts peut réaliser certaines opérations de calcul logique. Ainsi, on peut réaliser un réseau de neurones formels permettant d'effectuer n'importe quel calcul, et équivalent à une machine de Turing ou à des automates finis. En revanche, un seul neurone ou même un réseau mono-couche tel que le Perceptron ne peut pas réaliser d'opérations non linéaires telles qu'un ou exclusif, comme l'ont prouvé Minsky et Papert [Minsky and Papert, 1968]. Suite à leur publication, dont les implications n'avaient pas été bien comprises par tous, le connexionnisme a été temporairement abandonné par les financements publics. Comme l'a expliqué Minsky par la suite :

« Ce qu'on prouve dans ce livre, c'est que dans [les perceptrons classiques], il n'est pas possible de calculer certains prédicats à moins qu'au moins un des neurones ne soit connecté avec un poids non nul à chaque entrée. Ceci allait à l'encontre de ce qu'espéraient certains chercheurs, qu'on pourrait essentiellement se contenter de

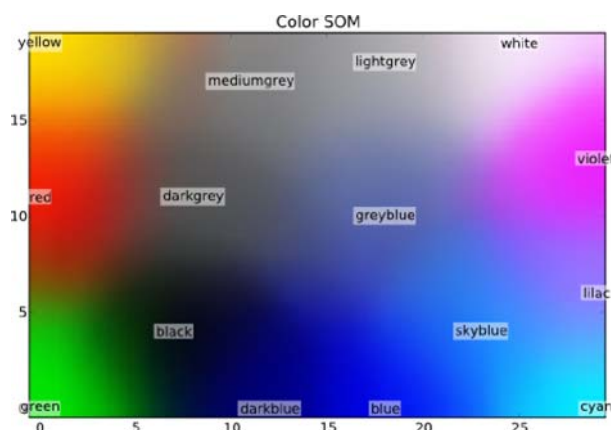


FIGURE 4.5 – Classement de couleurs par une carte auto-adaptative

réseaux avec quelques couches de neurones “locaux”, chacun n’étant connecté qu’à un petit nombre d’entrées. Un réseau “feed-forward” et avec des neurones locaux est beaucoup plus simple à construire et à utiliser qu’un réseau de neurones plus grand et récurrent, et dont les chercheurs, à l’époque, se concentraient sur ceux-là et non sur des réseaux plus compliqués. »

L’Adaline était soumise à la même limitation théorique que le Perceptron, à savoir, ne pouvant traiter que les problèmes linéairement séparables.

Afin de résoudre la limitation aux problèmes linéairement séparables ont été inventés les réseaux multicouches. Il a par la suite été prouvé qu’un Perceptron multicouche pouvait réaliser l’approximation uniforme de toute fonction bornée suffisamment régulière avec une précision arbitraire. Toutefois, aucun résultat théorique n’a pu être obtenu sur la conception topologique optimale des réseaux pour une classe de problèmes donnée.

4.2.3 Apprentissage profond

Les algorithmes que nous avons vus jusqu’ici sont bien adaptés à l’apprentissage de modèles simples, mais quand les variables sous-jacentes au problème sont en fait cachées, il faut employer d’autres méthodes. Ainsi, l’apprentissage profond, ou *deep learning* en anglais, est une discipline qui tente de modéliser des abstractions de haut niveau par rapport aux données fournies, par le biais d’architectures employant de multiples transformations non linéaires.

De nombreux algorithmes d’apprentissage automatique utilisent des représentations de données distribuées. Ces représentations correspondent à l’hypothèse que les données sont générées par l’interaction de différents facteurs. L’apprentissage profond rajoute l’hypothèse que ces différents facteurs sont organisés en une hiérarchie de plusieurs niveaux. Ces facteurs peuvent se composer entre eux ou correspondre à différents niveaux d’abstraction.

Lorsque les causes des données sont structurées ainsi, une architecture peu

profonde risque alors de ne pas parvenir à capturer les causes de plus haut niveau. Ainsi, Andrew Ng argumente que dans les travaux d'Olshausen & Field [Olshausen et al., 1996], on peut penser que c'est la faible profondeur de leur architecture qui fait que les invariances détectées étaient typiquement d'assez bas niveau, correspondant à des détections de contour.

En conséquence, les algorithmes d'apprentissage profond exploitent cette hypothèse en utilisant des représentations distribuées à plusieurs niveaux. En rajoutant un niveau, on peut accéder à davantage d'abstraction, ou déceler une nouvelle cause. Souvent, les algorithmes construisent d'abord la représentation du niveau le plus bas, et progressent dans la hiérarchie, même si certains algorithmes utilisent un apprentissage parallèle de tous les niveaux. Dans tous les cas, c'est la construction des concepts de bas niveau qui permet d'apprendre les concepts de haut niveau, en démêlant partiellement l'écheveau des causes. Il est souvent possible également de faire cet apprentissage de façon non supervisée, ce qui est un gros avantage.

La notion de profondeur d'un algorithme d'apprentissage peut être formalisée comme étant la profondeur de son graphe de flot de contrôle, où chaque opération élémentaire est représentée par un nœud. La profondeur d'un tel graphe est alors définie comme la longueur du plus long chemin depuis une entrée jusqu'à une sortie.

La profondeur d'un réseau de neurones sans récurrence peut être considérée comme égale au nombre de couches (c'est-à-dire le nombre de couches cachées plus la couche de sortie). Les machines à vecteurs de support (*support vector machines*, ou SVM) ou les perceptrons ont une profondeur de 2.

Les motivations principales pour étudier les algorithmes d'apprentissage pour des architectures profondes sont les suivantes :

- Le cerveau a une architecture profonde ;
- Les représentations et processus cognitifs semblent également être profonds ;
- Les architectures profondes sont plus efficaces.

Architecture profonde du cerveau Le cerveau est structuré en aires fonctionnelles (plus de détails sont disponibles en annexe). Par exemple, pour la vision, l'information brute en provenance du nerf optique entre par l'arrière du cerveau dans l'aire visuelle primaire, qui permet de dégager des signaux très basiques, comme de petits segments, de petits mouvements, la couleur, le contraste et la différence entre ce que voient les deux yeux ; cependant, à cause des saccades oculaires, son contenu varie très vite. Les aires suivantes vont pouvoir se baser sur ces informations pour répondre à des concepts plus abstraits. Ainsi, vers la fin du processus, dans les aires plus avancées, on peut trouver des neurones correspondant à des formes basiques comme une étoile bleue, ou capables de reconnaître objets et visages, et de gérer la notion de mouvement des objets. Ces informations traitées sont ensuite communiquées au cortex associatif, qui fait le lien entre les différents sens.

Nous voyons donc que les aires du cerveau fonctionnent bien en hiérarchie, même si ce n'est pas une hiérarchie pure, puisque des aires de niveau éloignés sont reliées entre elles et que plusieurs chemins parallèles peuvent coexister. On peut donc voir

le néocortex comme une hiérarchie de filtres, où chaque couche capture une partie de l'information qu'elle reçoit, la traite, et transmet cette information traitée ainsi que tout ce qu'elle n'a pas pu traiter aux couches situées plus haut dans la hiérarchie. Daniel Felleman et David van Essen ont ainsi pu dresser la cartographie hiérarchique des aires du cortex des primates [Felleman and Van Essen, 1991], qui est présentée dans la figure 4.6¹.

En outre, l'apprentissage profond correspond également à certaines théories de développement du néocortex. Selon ces modèles, le cerveau déclenche diverses vagues de facteur de croissance neurotrophique, qui modifie la structure du cerveau en développement, en reliant de nouvelles régions aux régions déjà structurées et matures. Le cerveau humain est en effet unique parmi les primates pour sa plasticité tardive, qui peut donc permettre à la complexité du monde de structurer le cerveau, contrairement aux autres animaux pour qui cette structuration sera principalement d'origine évolutive, ce qui explique notre adaptabilité unique [Elman, 1998].

Profondeur des représentations et processus cognitifs On peut remarquer que les humains organisent leurs idées et leurs concepts de façon hiérarchique. Ainsi, nous avons le concept assez basique de chien, qui est une abstraction composée à partir de différents exemples de chiens. Cependant, en creusant un peu, on peut en reconnaître des sous-catégories correspondant aux diverses races de chien (caniche, chihuahua, labrador...), tout comme on peut également reconnaître que ce concept fait partie de plusieurs catégories plus abstraites : un chien est aussi un animal domestique, un mammifère, un vertébré, un animal, un être vivant.

On peut supposer que cette structuration de nos concepts vient d'un apprentissage progressif des concepts, où les humains apprennent d'abord des concepts simples, et les combinent ensuite pour représenter des concepts plus abstraits. Ainsi, un enfant va d'abord voir trois caniches, un berger allemand, cinq bâtards et un dalmatien, avant de pouvoir constituer les concepts correspondant à chacune de ces différentes races, mais également le concept de ce qu'est un chien en général, en mettant en commun leurs caractéristiques communes et en faisant abstraction de leurs différences.

Cette forme de représentation permet un transfert d'apprentissage plus poussé. Ainsi, apprendre un fait général sur les chiens (par exemple qu'ils peuvent mordre quand ils ont peur) permet d'étendre automatiquement cette connaissance à tous les sous-concepts dépendant de cette catégorie (les caniches, le chihuahua de la voisine), et de supposer qu'elle peut également s'étendre à certaines autres catégories dépendant de la même catégorie supérieure (pour poursuivre notre exemple, d'autres animaux).

Par ailleurs, on en déduit également des règles plus générales, qu'on va utiliser comme contraintes implicites par la suite. Ainsi, à force d'étudier différents animaux, on peut déduire qu'au sein d'une même espèce, tous les individus ont le même nombre d'yeux et de pattes. Ce type d'inférence s'appelle une surhypothèse.

1. Source : [Felleman and Van Essen, 1991]

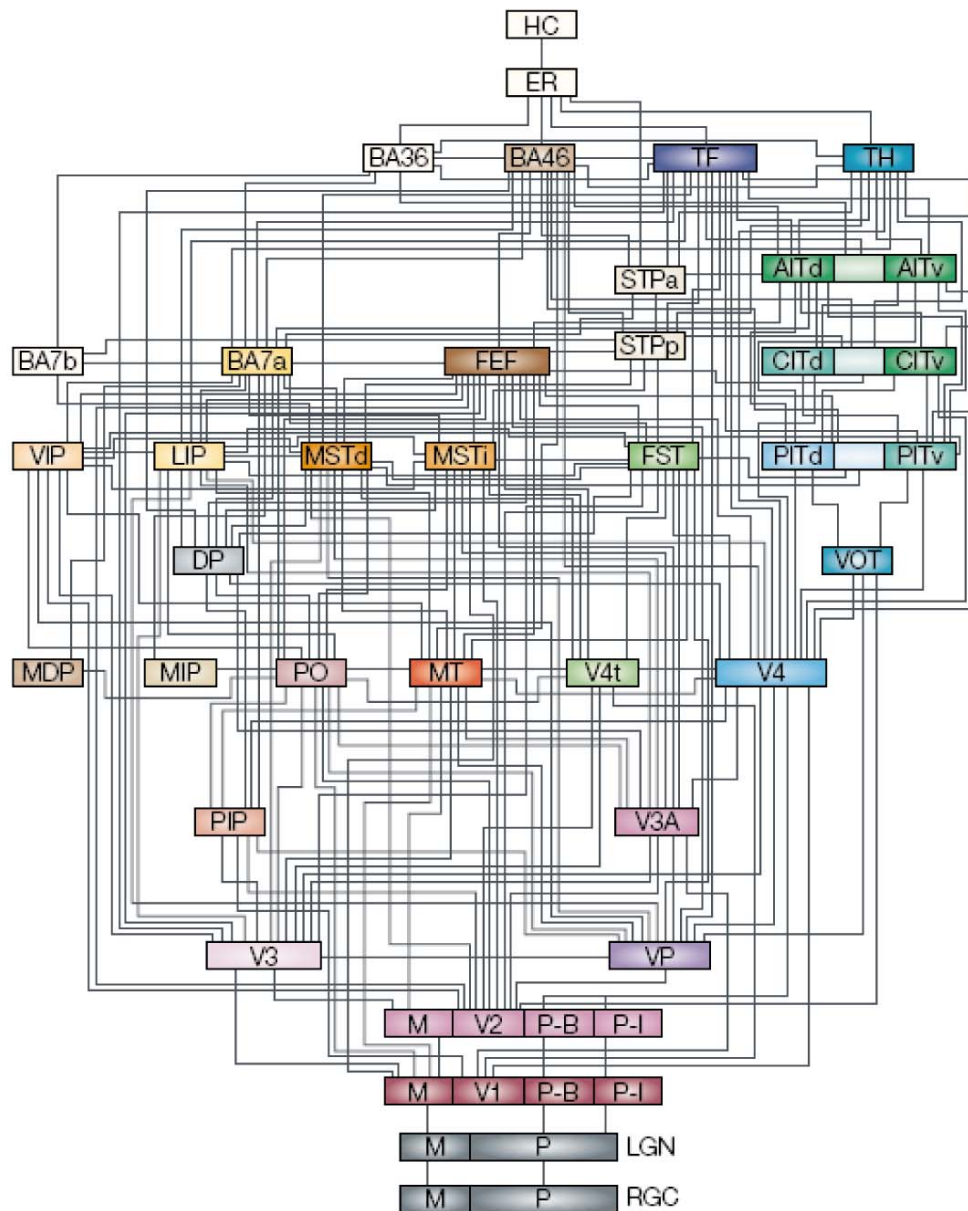


FIGURE 4.6 – Cartographie hiérarchique des aires du cortex des primates établie par Felleman et Van Essen

Efficacité des architectures profondes Nous avons indiqué dans la partie précédente des résultats théoriques obtenus sur les réseaux de neurones visant à prouver que certains réseaux peuvent approximer uniformément toute fonction bornée suffisamment régulière avec une précision arbitraire. C'est notamment le cas des perceptrons multicouches, dont l'architecture est profonde. C'est en fait également le cas de

certaines réseaux qui ont seulement une profondeur de 2 : ce résultat est valable pour les portes logiques, les neurones formels (à seuil), les neurones avec une fonction d'activation sigmoïdale, et les unités à base radiale comme dans les SVMs.

Mais cette possibilité a un prix : le nombre de nœuds requis dans le graphe peut devenir très large. En fait, des résultats théoriques indiquent que, pour certaines familles de fonctions, le nombre de nœuds nécessaires peut grandir selon l'exponentielle du nombre de dimensions des entrées. Cela a été montré pour les portes logiques, les neurones formels, ainsi que les neurones à base radiale (RBF). Dans ce dernier cas, Hastad a montré que certaines familles de fonctions, qui peuvent être représentées efficacement (de façon compacte) avec $O(n)$ nœuds (pour n entrées) lorsque la profondeur est d , ont besoin d'un nombre exponentiel de nœuds ($O(2^n)$) lorsque la profondeur est limitée à $d-1$.

Il s'agit là bien entendu d'un problème de mémoire, mais également d'un problème de temps de calcul. Non seulement chaque étape de l'algorithme doit être répercutée sur davantage de nœuds, mais, comme chacun de ces nœuds doit être entraîné aussi longtemps, il faut présenter davantage d'exemples.

Une architecture profonde peut donc être vue comme une forme de factorisation, permettant d'avoir moins d'unités à entraîner. Beaucoup de fonctions qui peuvent être représentées efficacement par une architecture profonde ne peuvent pas l'être avec une architecture peu profonde. L'existence d'une représentation compacte et profonde indique qu'il existe une certaine structure dans la fonction représentée. Si la fonction n'a pas de structure du tout, il sera de toute façon impossible de la représenter.

Pour comprendre cette factorisation, nous devons expliquer comment les représentations intermédiaires peuvent être exploitées pour partager la force statistique, en mettant à contribution différents contextes. Il s'agit en fait de réutiliser une unité donnée de l'architecture dans différents contextes. Cette réutilisation permet d'utiliser plus d'information pour estimer les paramètres de cette unité : on a donc besoin de moins d'exemples à présenter. Ce partage est similaire à celui qui a lieu dans l'apprentissage de représentations distribuées et qui fait leur force, permettant d'identifier partiellement une nouvelle donnée par ses composantes connues.

Le partage de la force statistique a produit de nombreuses avancées dans le domaine de l'apprentissage automatique, notamment dans l'apprentissage multi-tâches, qui l'exploite particulièrement bien et où il produit de très bons résultats. [Collobert and Weston, 2008]

Prenons un exemple où l'on cherche à se former une représentation mentale du contenu de plusieurs sacs de billes, dont la répartition est identique : chacun contient deux fois plus de boules rouges que de boules bleues. On peut les observer uniquement en faisant des tirage aléatoires consistant à prélever une bille dans un sac.

Si l'on cherche à les modéliser, par exemple avec des modèles génératifs hiérarchiques, on observera vite que proposer un niveau d'abstraction supplémentaire, correspondant à un prototype commun pour chaque sac, accélère l'apprentissage, dans la mesure où les sacs correspondent effectivement à ce prototype commun. Dans ce sens, le partage de force hiérarchique signifie que chaque tirage fournit aussi une

indication pour tous les autres sacs, certes plus faible qu'un tirage les concernant, mais significative. Cette indication est également valable pour des sacs qui n'ont jamais été vus. Cela correspond au phénomène cognitif du transfert d'apprentissage. Dans le domaine de l'apprentissage automatique, c'est ce qu'on appelle le transfert inductif, ou l'apprentissage par transfert, qui est souvent réalisé à partir de réseaux bayésiens ou de réseaux de logique probabiliste.

Nous avons parlé dans le début de ce chapitre de la malédiction de la dimensionnalité, si présente en apprentissage automatique. Mais nous voyons ici que si l'ajout de nouveaux paramètres correspond à l'ajout de niveaux d'abstraction du modèle, il devient au contraire possible d'apprendre davantage avec moins d'observations. On peut donc parler à ce propos de bénédiction de l'abstraction [Goodman, 1983]. Cette bénédiction peut sembler contre-intuitive dans la mesure où le modèle est rendu plus complexe, mais au contraire, cette complexité ajoute en fait des contraintes, dont il est bien connu qu'elles facilitent l'apprentissage. Elle permet notamment la modélisation des surhypothèses.

Algorithmes et résultats Différents algorithmes existent basés sur diverses autres technologies : on a ainsi des réseaux de neurones profonds, des réseaux de neurones à convolution profonds, et des réseaux de croyance profonds.

Les premiers exemples d'algorithmes à architecture profonde sont des réseaux de neurones artificiels profonds. Le premier est le Neocognitron de Kunihiko Fukushima, en 1980 [Fukushima, 1988]. Il se base sur les résultats de Hubel et Wiesel [Hubel and Wiesel, 1959, Hubel and Wiesel, 1962], qui avaient détecté dans le cortex visuel primaire deux types de cellules, les cellules simples et les cellules complexes, ces dernières étant capables d'invariance spatiale. Ainsi, le Neocognitron est constitué de cellules S, qui extraient les caractéristiques locales, et de cellules C, qui sont tolérantes aux déformations et déplacements de ces caractéristiques. Une cellule C est reliée en entrée à un ensemble de cellules S correspondant à la même caractéristique, mais à des positions différentes ; elle s'active si au moins une de ses entrées s'active. Elles réalisent donc un floutage spatial.

L'architecture du réseau alterne des couches de cellules S et de cellules C. Aux bas niveaux de la hiérarchie, les caractéristiques découvertes par les cellules S seront des caractéristiques de bas niveau : ainsi, pour une utilisation dans le domaine de la vision, cela pourra être la présence de bordure ou l'orientation d'une ligne. Dans les niveaux supérieurs, les caractéristiques reconnues seront plus abstraites et moins locales grâce au filtrage fait par les couches S précédentes et à l'intégration faite par les couches C précédentes. Cette architecture est présentée dans la figure 4.7.

L'apprentissage se fait de façon non supervisée. Des liaisons inhibitrices forcent l'activation d'une seule cellule S dans un rayon donné, et cette cellule apprend suivant une règle de Hebb qui va forcer sa spécialisation.

Le Neocognitron était en fait le premier exemple de réseau de neurones à convolution. On désigne par ce terme les réseaux constitués de neurones qui répondent à des zones locales, avec recouvrement, d'une image plus globale à deux dimensions.

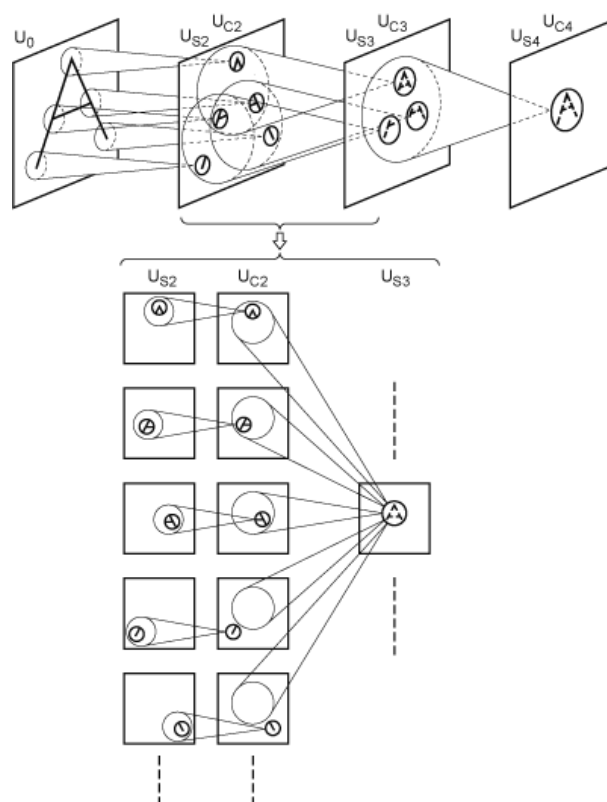


FIGURE 4.7 – Neocognitron

Ces réseaux ont nécessairement une architecture profonde pour parvenir à une vision d'ensemble. En raison de leur tolérance aux opérations de translation, ils sont bien adaptés à la reconnaissance visuelle. Un autre exemple d'algorithme relevant de cette catégorie est le HMAX de Riesenhuber et Poggio [Riesenhuber and Poggio, 1999].

En 1989, Yann Le Cun a entraîné un modèle semblable avec une rétropropagation supervisée [LeCun et al., 1989]. Ces modèles supervisés ont eu peu de succès, pour une raison identifiée en 1991 par Sepp Hochreiter [Hochreiter, 1991, Hochreiter, 1998] comme le problème de disparition ou d'explosion du gradient, qui affecte aussi les réseaux de neurones récurrents, quand ils sont entraînés par rétropropagation profonde. Une erreur se propageant de couche en couche va croître de façon exponentielle par rapport au nombre de couches (c'est l'explosion), ou au contraire rapidement diminuer au point de ne plus avoir aucun impact après quelques couches (c'est la disparition).

Pour résoudre ce problème, plusieurs solutions vont être proposées. La première est celle de Jürgen Schmidhuber, en 1992, qui entraîne les niveaux un par un de façon non supervisée mais avec rétropropagation [Schmidhuber, 1992]. La représentation précédemment apprise par une couche est donc utilisée comme entrée par la couche suivante. Une deuxième est celle de Hochreiter et Schmidhuber, qui proposent en 1997 leur LSTM (long short term memory) [Hochreiter and Schmidhuber, 1997].

Sven Behnke choisit de se baser uniquement sur le signe du gradient, et non sur sa valeur, pour son modèle Neural Abstraction Pyramid [Behnke and Rojas, 1998]. Enfin, les méthodes aléatoires, par exemple les méthodes évolutionnaires, permettent une exploration suffisamment efficace de l'espace d'état des poids de connexion pour pouvoir suffire. Par ailleurs, ce problème tend maintenant à disparaître par simple augmentation de la puissance de calcul des ordinateurs.

En 2006, le domaine a été bouleversé par les travaux de Geoffrey Hinton sur les réseaux de croyance profonds (deep belief networks) [Hinton et al., 2006]. Dans son premier modèle, chaque couche est en fait une machine de Boltzmann restreinte. De Schmidhuber, il garde l'apprentissage non supervisé, niveau par niveau. Cependant, il le fait suivre d'un apprentissage supervisé par rétropropagation qui viendra raffiner ce pré-entraînement, notamment pour la couche de sortie, mais également potentiellement pour toutes les autres. Les résultats obtenus sont suffisamment bons pour susciter un nouvel intérêt pour ces architectures. En effet, chaque couche rajoutée augmente la borne inférieure de la probabilité logarithmique des données, ce qui montre que le modèle s'améliore.

Par la suite, Ranzato utilise des auto-encodeurs épars à convolution [Ranzato et al., 2007]. En fait, les réseaux de croyance profonds peuvent avoir comme couche n'importe quel modèle génératif, et sont surmontés par deux couches formant une mémoire associative. Dans ce modèle, les connexions vont dans les deux sens.

En 2011, Google a monté un projet surnommé Google Brain [Le et al., 2012], en partenariat avec l'université de Stanford, mené par Andrew Ng et Jeff Dean, d'apprentissage non supervisé par réseau de neurones profond contenant 1 milliard de connexions avec un codage clairsemé et massivement distribué sur 16 000 cœurs de CPU. Le réseau utilisé a une structure complexe inspirée de Neocognitron et HMAX, utilisant trois couches : un filtre, un pooling local, et une normalisation de contraste. L'apprentissage se fait selon une règle de Reconstruction Topographic Independent Component Analysis [Hyvärinen et al., 2009]. Après avoir reçu comme entrée un échantillon de 10 millions d'images issues de vidéos YouTube par sélection aléatoire, le réseau a appris à reconnaître des concepts de haut niveau tout en étant robustes aux invariances (translation, mise à l'échelle, rotation hors du plan). Les trois catégories les mieux reconnues, correspondant aux concepts occurring le plus souvent dans les vidéos postées sur Internet, étaient des visages humains, des silhouettes humaines et des chats. La performance de ce détecteur sur le benchmark ImageNet était de 15,8%, soit une amélioration de 70% par rapport à l'état de l'art.

Dans divers domaines tels que la vision par ordinateur, la reconnaissance de langage parlé ou écrit, ou la reconnaissance de musique ou d'autre signal audio, ces techniques ont produit des résultats au niveau des meilleurs autres algorithmes. Les algorithmes d'apprentissage profond supervisé ont même été les premiers à atteindre des performances comparables à celles des humains dans certains domaines de reconnaissance de motifs. Actuellement, ce sont les réseaux de neurones à convolution qui s'avèrent les plus performants, notamment dans le domaine de la vision, sur les benchmarks MNIST et NORB.

Des travaux récents [Bengio, 2009] ont montré que des architectures profondes

bénéficient plus de l'apprentissage autodidacte et de l'apprentissage multi-tâches que les architectures peu profondes. Elles sont également très appropriées pour la généralisation hors-domaine, notamment en ce qui concerne la reconnaissance de motifs et pour le traitement du langage naturel (natural language processing, NLP), plus particulièrement l'analyse de sentiment [Glorot et al., 2011].

Toutefois, des travaux récents viennent jeter un discrédit certain sur les réseaux de neurones profonds. En effet, [Szegedy et al., 2013] ont montré que dans plusieurs réseaux de neurones profonds distincts, et pour plusieurs jeux de données distincts sur lesquels ces réseaux avaient été entraînés à réaliser des tâches de classification, il leur était possible de prendre une image correctement classifiée et de lui apporter une modification qui, bien qu'invisible à l'œil nu par un humain, suffisait à provoquer une erreur de classification. Leurs résultats semblent en outre montrer une absence de sémantique des unités.

Ces problèmes nous semblent importants malgré les bons résultats obtenus par les architectures profondes. Nous pensons toutefois que HTM-CLA y est beaucoup moins sensible du fait de la sémantique importante apportée par la dimension temporelle, ce qui justifie encore davantage notre choix.

4.2.4 Cortical Learning Algorithm, un paradigme de compréhension et d'imitation du néocortex

Après avoir justifié notre intérêt pour HTM-CLA en première partie de ce chapitre, et avoir ici présenté les domaines voisins que sont la biologie du cerveau, les réseaux de neurones artificiels et l'apprentissage profond, nous pouvons maintenant nous appesantir sur une description un peu plus poussée de cet algorithme et de ses spécificités.

4.2.4.1 Structure du néocortex

Comme nous l'avons vu plus haut, le néocortex a une structure uniforme. Rien ne permet de différencier les aires du cerveau au microscope ; seules leurs fonctions diffèrent. Cette réalité biologique correspond à la structure corticale proposée dans ce modèle, selon lequel le néocortex applique un seul algorithme d'apprentissage de façon massivement parallèle et obtient en retour une représentation des motifs toujours semblable. Aucune spécialisation structurelle ne différencie l'encodage de la vision de celui de l'audition, ni de celui d'un concept abstrait. La seule différence est qu'une des zones va être reliée au nerf optique, l'autre au nerf auditif et la dernière sera dans une des zones dites associatives, qui ne sont ni sensorielles, ni motrices, mais qui font le lien entre les autres zones et permettent justement ainsi l'émergence d'une pensée abstraite.

Le néocortex repère les motifs qui se répètent parmi ce que nos sens lui font parvenir. Un motif est défini comme étant la coïncidence temporelle d'un certain nombre de perceptions. Si cette coïncidence ne se reproduit pas, on peut considérer qu'elle était due au hasard. Si au contraire, elle revient souvent, c'est qu'elle doit

correspondre à une réalité du monde, c'est-à-dire à un invariant. Ce motif peut alors être « nommé » : un concept est formé qui en relie toutes les composantes. Cette notion de nommage a un réel fondement biologique, avec l'existence des neurones grand-mère² et des neurones miroir³. Une fois appris, un motif sera ensuite reconnu à chaque fois que ses composantes seront perçues.

Le nommage permet également l'organisation des motifs en hiérarchie, ce qui correspond à la structure naturelle du monde et facilite d'autant l'apprentissage. Ainsi, il est plus facile de mémoriser la notion de visage si l'on a déjà mémorisé les notions d'œil, de nez et de bouche ; il est plus facile d'acquérir le concept d'humain si l'on a déjà acquis les concepts de tête, de bras et de jambes ; et ainsi de suite, dans une hiérarchie qui va du pixel coloré perçu par un cône de notre œil jusqu'aux concepts abstraits.

4.2.4.2 Apprentissage des séquences temporelles

Les invariants que le cerveau a pour but de capturer sont en fait toujours temporels. Cela peut paraître évident lorsqu'il s'agit de mémoriser un déplacement ou une chanson, qui sont indissociables de leur structure temporelle. Mais de façon plus étonnante, notre façon de mémoriser impose ce biais même à des concepts qui pourraient sembler purement spatiaux. Ce fait est confirmé par diverses expériences de psychologie qui montrent que nos cartes mentales des lieux sont fondamentalement basées sur notre représentation de nos déplacements à travers ces lieux.

En conséquence, en plus de mémoriser les motifs qui se répètent, le néocortex mémorise également la façon dont ils s'enchaînent les uns avec les autres. Cette mémorisation se fait dans un premier temps sur des séquences extrêmement courtes, mais au fur et à mesure de leur consolidation dans la mémoire, elles pourront être regroupées en séquences plus longues. Chaque séquence est également nommée et inscrite dans la hiérarchie temporelle des connaissances.

C'est ce nommage qui provoque le phénomène d'invariance de la représentation. Considérons deux niveaux d'une hiérarchie en train de regarder un chien courir. Le niveau inférieur pourrait représenter les différents pixels correspondant au chien tels qu'ils apparaissent sur la rétine, le supérieur contiendrait l'étiquette de la séquence « chien qui court ». Au niveau inférieur, les motifs reconnus varient à chaque déplacement de notre chien : non seulement la position relative de ses pattes par rapport à son corps varie, mais en plus sa position absolue change également si l'observateur ne le suit pas du regard ; bref, il est tout à fait possible que presque aucun pixel ne soit en commun d'un instant à l'autre. En revanche, au niveau supérieur, le motif de la séquence « chien qui court » reste activé pendant toute la durée de la scène. Cette représentation à base de nommage hiérarchique de séquences temporelles a donc permis de faire apparaître une invariance. Plus on monte dans la hiérarchie des

2. Des neurones associés à un individu particulier. Plus de détails en annexe C.

3. Des neurones empathiques qui s'activent aussi bien quand leur propriétaire effectue une action que quand il observe un autre individu effectuer cette même action, voire même lorsqu'il imagine cette action. Plus de détails en annexe C.

représentations, plus les causes sous-jacentes aux invariants sont fondamentales, et donc plus l'invariant va durer longtemps.

Cette invariance est absolument nécessaire à la capacité de reconnaissance des motifs dont nous faisons preuve. Sans elle, il n'y aurait justement pas moyen de comprendre que ces deux photos représentent bien une même personne : c'est juste que sur l'une, elle est plus éloignée, donc apparaît plus petite, elle n'est pas vue sous le même angle, ni dans le même éclairage. L'importance de comprendre comment le cerveau parvient à une représentation invariante a été souligné dans [Pinto et al., 2008, DiCarlo et al., 2012]. Wallis et Rolls [Wallis and Rolls, 1997] ont montré comment des processus biologiques pouvaient donner naissance à ces invariances, et plusieurs modèles de réseaux de neurones ont eu pour but de les reproduire [Földiák, 1991, Ullman and Bart, 2004].

Un système connexionniste d'apprentissage et de reconnaissance hiérarchique de séquences spatio-temporelles a également été proposé dans [Nguyen et al., 2012].

En pratique, la continuité et la contiguïté temporelles jouent donc le rôle de supervision de l'apprentissage, en indiquant où se situe l'invariance, et en contraignant la représentation corticale à correspondre à cette invariance.

Behnke propose Neural Abstraction Pyramid [Behnke and Rojas, 1998], un réseau de neurones hiérarchique adapté à la vision, avec une abstraction croissante, comportant à la fois des liens directs et de rétroaction⁴ entre les étages de la hiérarchie. Il l'a utilisé pour numériser un texte manuscrit, mais après beaucoup d'adaptation manuelle, consistant à définir ce que devait contenir chaque couche : la plus basse devait séparer l'arrière-plan de l'écriture, le niveau suivant détecter les bords et le plus haut les agréger en lignes. Il prévoyait pour la suite le rajout d'un mécanisme d'attention.

Rolls s'est également intéressé [Rolls and Milward, 2000] à la création de l'invariance dans le cerveau et a proposé qu'elle était rendue possible par une hiérarchie de réseaux de neurones comportant tous une inhibition locale, où chaque neurone n'est connecté qu'à un champ réceptif assez local de la région inférieure, et utilisant une règle de Hebb modifiée pour conserver une trace de l'activité précédente du neurone. Ainsi, on peut utiliser ici aussi le temps comme superviseur, et le champ réceptif est également croissant quand on monte dans la hiérarchie.

À partir de cela, Stringer et Rolls [Stringer and Rolls, 2002] ont conçu un modèle du circuit ventral de la vision, nommé VisNet, comportant quatre régions en hiérarchie, censées correspondre respectivement à V2, V4, au cortex temporal postérieur inférieur et au cortex temporal antérieur inférieur. Toutefois, ils n'ont prévu que des connexions de type feedforward. La trace de l'activité précédente du neurone a pris la forme d'une moyenne temporelle pondérée de son activité. Ils ont tout de même noté le développement d'invariances de types translation, changement de taille et d'orientation dans leur réseau.

Miller et Lommel ont utilisé des cartes auto-organisatrices (SOM en anglais) pour construire un modèle assez proche du paradigme mémoire-prédiction, Hierarchical

4. Les liaisons directes vont de l'entrée vers la sortie du réseau, à l'inverse de celles de rétroaction.

Quilted SOM (HQSOM) [Miller and Lommel, 2006], s’inspirant de VisNet et HTM. Ils utilisent pour cela des paires de SOM classiques, qui réalisent le regroupement spatial (ou qu’on peut comparer aux neurones simples), et de cartes auto-organisatrices récurrentes (RSOM), qui sont capables de réaliser le regroupement temporel (analogues aux neurones complexes). Ils l’adaptent en n’autorisant qu’un seul neurone à s’activer, et n’utilisent que des liaisons feedforward. Pinto [Pinto and Engel, 2009] a étendu ce modèle pour se rapprocher encore de HTM, avec LoopSOM, qui rajoute les connexions feedback et permet donc une meilleure reconnaissance de motifs en présence de bruit ou d’ambiguïté. Il propose également l’étude de systèmes basés sur d’autres SOM, comme AntSOM [Swarup et al., 2005a], nativement capable de prédiction, ou RecAntSOM [Swarup et al., 2005b], son équivalent récurrent.

Ainsi, par l’apprentissage de ces séquences, le cortex se comporte de façon à construire des explications aux entrées qui lui parviennent [Gregory, 1980]. Il se comporte donc comme un modèle de génération [Dayan et al., 1995].

4.2.4.3 Intelligence par la prédiction

La mémorisation de séquences de motifs permet au néocortex de prédire en permanence ce qui devrait se produire à l’instant suivant et donc de remarquer tout de suite si quelque chose d’imprévu arrive.

Cette capacité biologique est confirmée par de récentes découvertes, comme dans une étude centrée sur le cortex visuel primaire, qui a montré une plasticité dans cette aire répondant à la présentation répétée d’une séquence toujours identique [Gavornik and Bear, 2014]. Après apprentissage, l’évocation du début de la séquence évoquait également l’activation de la fin de la séquence, ce qui montre également les capacités de prédiction du cortex même dans ce bas niveau de la hiérarchie corticale. En outre, une autre étude a montré que dans le chemin visuel ventral, les réponses des neurones dépendaient bien davantage des attentes et de l’éventuelle surprise perçue que des stimuli réellement observés [Egner et al., 2010].

Selon Hawkins, cette prédiction se fait en parallèle à tous les étages de la hiérarchie, de façon descendante. Elle se base à la fois sur l’état actuel et sur les connaissances acquises. À chaque étage, si un invariant prévoit d’être maintenu à l’instant suivant, il en informe le niveau inférieur, qui prévoit alors la prochaine étape de la séquence correspondant à cet invariant. De proche en proche, on en arrive à des prévisions de plus en plus détaillées, mais qui ont de moins en moins de chances de se produire avec précision. Ces prévisions se font également à terme variable : si les prévisions de très bas niveau sont forcément très limitées dans le temps, et ne sont probablement valables que pour l’instant suivant, les prévisions des étages supérieurs peuvent être valables pour une assez longue période de temps.

On voit donc qu’il existe dans le néocortex, selon ce modèle, un double circuit de l’information : les perceptions entrent par le bas de la hiérarchie et l’information remonte à mesure qu’une entrée est reconnue comme correspondant à un motif, puis l’information de prédiction redescend la hiérarchie.

Il arrive parfois que la prévision ne corresponde absolument pas à la perception.

Quand cela arrive, l'information de ce décalage est remontée plus haut dans la hiérarchie, permettant aux niveaux supérieurs de remettre en cause leur interprétation et d'en fournir une plus cohérente avec la situation réelle.

Les prévisions servent également à analyser les perceptions à un pas de temps donné. Elles fournissent un cadre qui peut permettre de distinguer entre des interprétations différentes de données ambiguës. C'est ce qui explique par exemple un certain nombre d'illusions d'optique qui peuvent être vue de deux façons différentes, mais pas des deux façons à la fois.

La majeure partie du temps, ce mécanisme de prédiction n'est absolument pas présent de façon consciente à notre esprit. Cependant, quand une prédiction s'avère erronée, cela provoque souvent une réaction consciente, sans parfois que l'on parvienne à analyser tout de suite la source de l'erreur. « Tu as quelque chose de changé... Une nouvelle coupe de cheveux? Non? De nouvelles lunettes? »

L'exemple choisi par Hawkins pour illustrer ce phénomène est l'ouverture d'une porte dont nous sommes familiers, par exemple la porte de notre chambre. Selon une représentation cognitiviste, nous devrions avoir en mémoire toutes sortes d'attributs reliés à cette porte: sa hauteur, sa largeur, sa couleur, sa texture... Imaginons maintenant qu'un plaisantin ait réussi, sans modifier l'apparence de la porte, à en déplacer la poignée de dix centimètres vers le bas. Nous approchons de cette porte en mode pilote automatique, avançons la main et... dès l'instant où la main ne rencontre que le vide, au lieu de la poignée attendue, nous sursautons: que se passe-t-il? Selon Hawkins, ce que nous pourrions ressentir à ce moment-là est la surprise due à une prédiction erronée qui force son objet à subir un nouvel examen conscient. En comparaison, selon la représentation cognitiviste, il aurait fallu, à tout instant, alors que nous nous approchons de la porte, passer en revue toutes ses caractéristiques pour vérifier qu'elles n'avaient pas changé, ce qui induirait une surcharge cognitive; en outre, ce modèle n'explique pas pourquoi le changement de hauteur de la porte aurait été détecté si tardivement, et non pas par les yeux.

Ce mécanisme peut aussi fournir une explication partielle au mécanisme de l'attention involontaire: se porte à notre attention ce qui ne correspond pas à nos prédictions.

C'est également une forme de ce mécanisme de prédiction qui sert à résoudre les perceptions ambiguës, comme dans l'exemple des illusions d'optique. Une fois une hypothèse acceptée à haut niveau, elle se propage vers le bas de la hiérarchie, donnant un sens aux entrées précédemment ambiguës. Il faut alors faire un effort pour trouver une autre interprétation, et il est quasiment impossible de revenir dans l'état d'indécision qui précédait la première hypothèse formée. Un exemple de telle perception ambiguë est présentée dans la figure 4.8.

La prédiction est également ce qui permet d'anticiper comment l'action à venir va influencer sur l'environnement, et donc sur la perception au pas de temps suivant. Morse [Morse and Ziemke, 2009, Morse et al., 2010] a étudié l'Echo State Network (ESN), un réseau de neurones hiérarchique récurrent, qui utilise des cartes auto-organisatrices (SOM), pour montrer que le contrôle de l'action est nécessaire à la compréhension de la perception. Toutefois, il ne s'en sert pas pour la génération de comportement.



FIGURE 4.8 – Image très difficile à comprendre initialement, représentant un dalmatien dans un environnement très bruité

Selon ce modèle, c'est également ce mécanisme de prédiction qui fournirait la composante de commande motrice associée au comportement, tout en bas de la hiérarchie. Si un comportement à haut niveau est nécessairement abstrait (« aller dans la pièce d'à côté »), il est constitué à son échelle élémentaire d'une suite d'ordres à transmettre aux muscles (« contracte tel et tel muscle de la cuisse et du mollet »). Dans cette théorie, le mécanisme qui sert aux étages supérieurs de la hiérarchie à transmettre une prédiction sert également aux étages inférieurs à donner un ordre moteur. Les hiérarchies sensorielles et motrices sont nécessairement couplées, puisque toute action influe sur l'environnement, et donc sur les perceptions, et doit donc être prise en compte dans toute prédiction pour qu'elle soit valable.

Cette importance de la prédiction hiérarchique dans le fonctionnement du cortex a d'abord été élaborée par Srinivasan et al. en 1982 [Srinivasan et al., 1982]. Ils ont intitulé « codage prédictif » (predictive coding) leur modèle de fonctionnement du néocortex, par analogie avec des algorithmes de compression vidéo se basant sur la redondance des données. Cela correspond également aux hypothèses de Mumford sur les circuits cortico-corticaux [Mumford, 1992]. En 1999, Rao et Ballard en ont tiré [Rao and Ballard, 1999] un réseau de neurones dans lequel les connexions feed-forward ne servent effectivement plus qu'à transporter les erreurs résiduelles entre les prédictions et l'activité réellement observée. Ils soulignent l'efficacité de cette méthode d'encodage des données, qui ôte la redondance spatiale et temporelle naturellement présente dans les données. Ils montrent également la capacité de leur modèle à reproduire plusieurs comportements biologiquement observés, qui se confirme dans d'autres études [de Wit et al., 2010]. Leur modèle n'est cependant pas hiérarchique.

Schmidhuber a également reconnu l'intérêt d'un modèle prédictif dès 1991 ; qui plus est, il a également utilisé une architecture profonde dans son modèle, intitulé

Deep Learner [Schmidhuber, 1992], qui en fait donc l'un des plus proches de celui d'Hawkins. Il utilise un entraînement non supervisé, couche par couche, visant à permettre à chaque couche de prédire l'entrée à venir. Son modèle réalise ainsi une double compression spatiale et temporelle des données, et existe également dans une version à temps continu. Schmidhuber a même travaillé sur la compression de données en les faisant descendre dans la hiérarchie [Schmidhuber et al., 1993].

Dans l'argumentaire de Hawkins en faveur d'un fonctionnement fortement basé sur la mémoire, on retrouve l'argument de la rapidité de fonctionnement de nombreuses opérations cérébrales. En effet, la plupart de ces opérations, comme par exemple la classification visuelle, qui permet de décider si une image contient ou non un animal, se fait en une demi-seconde ou moins. Puisque, comme on l'a vu plus haut, un potentiel d'action met environ 5 millisecondes à se créer et se propager dans un neurone, cela signifie que ces opérations prennent moins de 100 « coups d'horloge » [Thorpe, 2009]. Les meilleurs algorithmes informatique actuels ne parviennent pas à faire cela avec une telle précision même en des milliards d'instructions, qui peuvent être ramenées à des millions avec une parallélisation du processus. Ainsi, l'hypothèse d'une forte utilisation permanente de la mémoire devient bien plus plausible, puisqu'une comparaison nécessiterait bien moins d'étapes qu'un calcul.

Cet argument du temps de calcul a pu être utilisé dans l'autre sens : ainsi DiCarlo [DiCarlo et al., 2012] considère que la reconnaissance visuelle en l'absence de forte ambiguïté se fait nécessairement par un chemin neuronal uniquement feed-forward. Toutefois, la possibilité de fonctionner plus rapidement dans ce mode feed-forward nous semble en fait être due à l'apprentissage qui a uniquement pu être fait avec feedback. Si le feedback n'est donc pas nécessaire à tout traitement, il reste néanmoins la base permettant l'apprentissage d'invariants par étiquetage constant. Hawkins insiste d'ailleurs sur le bien plus grand nombre de connexions cérébrales allant dans le sens du feedback que du feed-forward.

On peut faire une analogie entre ce modèle du cerveau et le fonctionnement d'une unité militaire. Chaque soldat individuel peut être considéré comme un neurone. Il récolte des informations de façon indépendante et confronte ses observations à ses prédictions. Il traite lui-même les phénomènes insignifiants, peut faire appel à des collègues pour traiter des phénomènes plus importants, mais surtout n'hésite pas à faire appel à son supérieur hiérarchique dès lors qu'il n'est plus capable de faire face seul. À son niveau, ledit supérieur hiérarchique fait de même, et en plus coordonne toute sa troupe. Il en est de même à tous les échelons.

La longueur du contexte temporel est également d'une grande importance. Plus un grand contexte temporel nous est fourni, plus la prédiction réalisée peut être précise. Comparons ainsi deux phrases auxquelles il manque le dernier mot :

« Qu'est-ce que ça peut bien vouloir __ ? »

« C'est __ ? »

Il est impossible de restreindre autant les possibilités de mots manquants pour la deuxième phrase que pour la première, par manque d'informations sur lesquelles se baser. Il est donc très important de pouvoir exploiter l'intégralité du contexte temporel fourni, par une mémorisation très détaillée à court terme et une bonne

mémorisation à long terme, pour pouvoir faire des prédictions le plus détaillées possibles.

En quoi ce mécanisme de prédiction continue peut-il être considéré comme une forme d'intelligence ? Premièrement, il permet un fonctionnement en pilote automatique dans notre environnement quotidien, ce qui a le mérite de traiter un grand nombre d'actions sans provoquer de surcharge cognitive. Deuxièmement, l'organisation des connaissances en hiérarchie permet de faire facilement des analogies, permettant d'aborder des situations nouvelles en se basant sur les situations connues, et créant un comportement flexible et adaptable. Un léger changement de la situation ne nécessite jamais une complète replanification. Troisièmement, il permet la faculté d'anticipation, qui est cruciale à l'intelligence, puisqu'elle permet d'évaluer les dangers, élaborer des plans et des stratégies, pousser une idée vers l'examen de ses conséquences, et plus généralement d'adapter son comportement non pas juste à l'instant présent, mais à ce qui va arriver dans le futur. C'est cette faculté d'anticipation qui a permis à l'être humain non seulement de pouvoir lutter contre ses prédateurs et capturer ses proies, mais également d'inventer l'agriculture.

4.2.4.4 Détails d'implémentation biologique

Nous avons mentionné précédemment que le néocortex était structuré en 6 couches et en colonnes, et que chaque colonne était une unité de base du fonctionnement massivement parallèle du néocortex. Selon le modèle mémoire-prédiction, une colonne serait donc une unité de mémorisation et de prédiction. Hawkins propose, suite à des études biologiques poussées, un circuit plus détaillé de l'information dans le néocortex, correspondant bien aux réalités biologiques qu'on trouvera légèrement plus détaillées en annexe C, et permettant d'explicitier ce fonctionnement dans un contexte de hiérarchie entre les colonnes [George and Hawkins, 2009]. Nous allons maintenant détailler ce circuit de l'information afin de voir à quelle partie correspond la proposition d'algorithme HTM-CLA.

Il existe deux circuits de l'information à travers les couches corticales : les circuits ascendant (voir figure 4.9) et descendant (voir figure 4.10). Les couches corticales sont traditionnellement numérotées depuis la surface du cortex vers sa profondeur. Nous allons dans un premier temps nous contenter de décrire ce circuit, puis nous citerons l'explication qu'en propose Hawkins.

Dans le circuit ascendant, l'information en provenance des régions inférieures dans la hiérarchie arrive dans les couches 4 et 6 d'une même colonne. La couche 4 relaie cette information en direction des couches 2 et 3 de cette même colonne. Ces couches relaient ensuite l'information dans les régions hiérarchiquement supérieures.

Le circuit descendant est un peu plus complexe. L'information entre par les longs axones qui composent la couche 1. Comme ces axones se prolongent sur de longues distances, cette information peut rentrer non pas dans une seule colonne mais dans plusieurs colonnes voisines simultanément. Elle y est captée par les neurones des couches 2, 3 et 5, qui y ont leurs dendrites. Les neurones des couches 2 et 3 relaient l'information vers les couches 5 et 6. Enfin, la couche 6 transmet l'information vers des

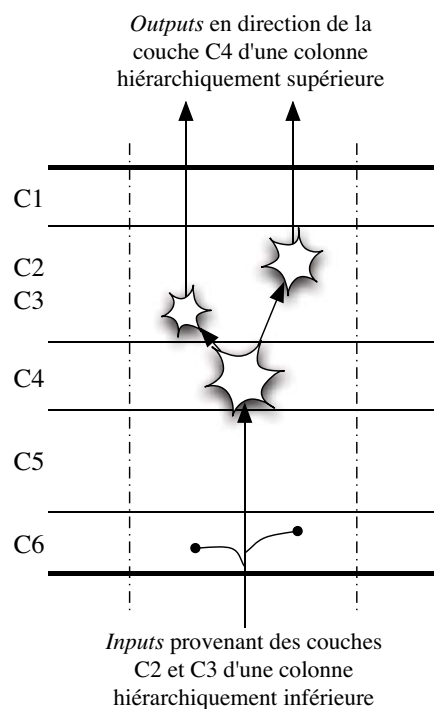


FIGURE 4.9 – Circuit ascendant de l'information

colonnes hiérarchiquement inférieures, par des axones qui aboutiront dans les couches 1 des régions visées. Quant à la couche 5, elle permet un mécanisme auto-associatif, que nous allons détailler.

Selon Hawkins, les neurones de la couche 5 de chaque colonne sont reliés aux cellules non-spécifiques du thalamus, une autre partie du cerveau, qui lui-même est relié en retour aux couches 1, non seulement de cette même colonne, mais également d'autres (voir figure 4.11). Si ces voies sont bien connues, la fonction du thalamus reste aujourd'hui encore assez confuse. L'hypothèse de Hawkins est que le rôle de cette structure est simplement de renvoyer vers les couches 1 les signaux qu'il reçoit des couches 5, occasionnant ainsi un léger retard. Ce retard permettrait de comparer à tout moment les informations actuelles avec les informations précédentes provenant de la même colonne, mais également d'autres colonnes.

La rétroaction ainsi constituée a deux rôles cruciaux dans le néocortex. Premièrement, il se trouve que les neurones de la couche 5 sont reliés aux neurones moteurs. Ce retour serait donc l'élément permettant une synchronisation et une coordination fines de tout mouvement. Deuxièmement, le retour dans une même région permet l'apprentissage et la reconnaissance des séquences, qui est possible grâce aux connexions provenant de la région supérieure, fournissant un contexte.

Pour illustrer ce concept, Hawkins utilise la métaphore de la chanson. Une chanson

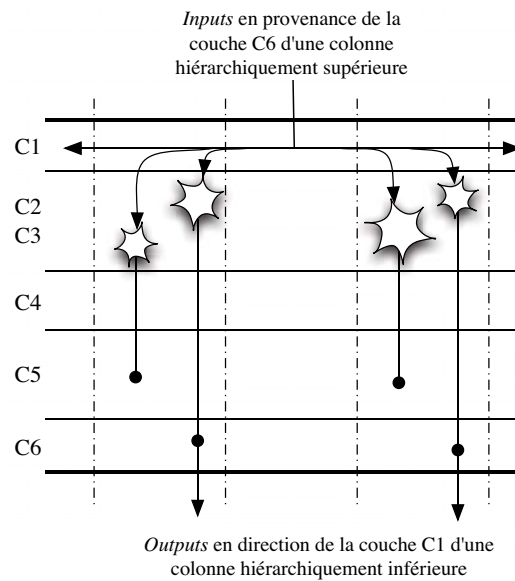


FIGURE 4.10 – Circuit descendant de l'information

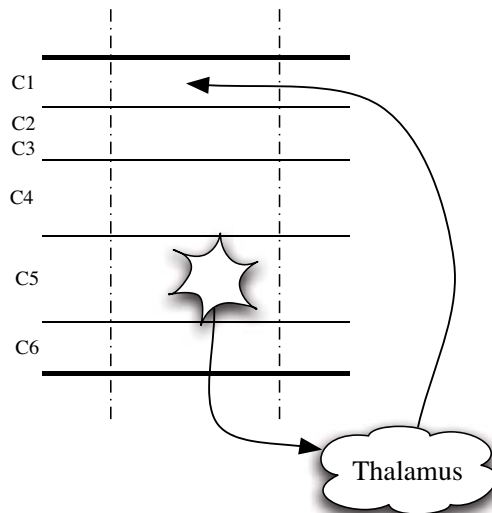


FIGURE 4.11 – Rôle du thalamus selon Hawkins

connue est une séquence temporelle de notes (ou plutôt d'écart entre notes, sauf pour ceux qui possèdent l'oreille absolue) associée aux méta-données de la chanson. Ici, la région supérieure fournit comme contexte le nom de la chanson (via la couche 6), indiquant qu'on est toujours en train de la chanter, et la région actuelle fournit un

contexte temporel (via la couche 5), constitué des cellules précédemment activées, qui indique où nous en sommes dans la chanson et également en quelle tonalité on la chante. C'est la réunion de ces deux informations qui permet d'activer la colonne correspondant à la prochaine note à chanter.

En temps normal, une colonne s'active quand les neurones de sa couche 4 s'activent, correspondant à une information provenant du bas de la hiérarchie, et donc des sens. Cependant, si cette activation correspond généralement à une activation dans la couche 1, ces synapses vont se renforcer par la règle de Hebb. Elles peuvent devenir suffisamment fortes pour qu'une activation provenant de la couche 1 suffise à activer la colonne. C'est ce qui correspond au mécanisme de prédiction. Puisque ce mécanisme vient de la couche 1, il vient indirectement de la combinaison de l'activation des couches hiérarchiquement supérieures et de l'activation retardée de la même région par le circuit thalamique. C'est cette combinaison qui permet d'utiliser un contexte générique appliqué à une situation spécifique pour faire une prévision spécifique, même si la situation n'avait jamais été rencontrée auparavant. C'est également ce qui permet d'interpréter des entrées floues : l'interprétation qui correspond le plus à la prédiction sera favorisée.

Afin de mettre en place le mécanisme de reconnaissance et de nommage de séquences ainsi que le mécanisme de détection d'anomalie, Hawkins propose le mécanisme suivant. Lorsqu'une séquence débute, la région supérieure en informe toutes les colonnes concernées via la couche 6, et ces colonnes activent toutes leur couche 2. Si toutefois l'activation de la couche 2 n'avait pas été prédite, une classe de neurones de la couche 3 s'allumerait en outre. Une autre classe de neurones de la couche 3 servirait à inhiber ces derniers dans le cas d'une prédiction correcte.

Enfin, il existe des connexions allant des neurones de la couche 6 d'une colonne aux neurones de la couche 4 de cette même colonne. L'utilisation de ce circuit correspond à l'utilisation des prédictions en tant qu'entrée : c'est le mécanisme de l'imagination et du rêve.

L'hypothèse de Hawkins propose également une interprétation du rôle crucial de l'hippocampe dans la formation de nouveaux souvenirs. Selon lui, l'hippocampe serait en fait l'équivalent d'une région supplémentaire située au sommet de la hiérarchie. Les seules informations qui l'atteignent sont les informations réellement nouvelles, qui ont remonté toute la hiérarchie. Sa structuration plus simple serait due essentiellement à son rôle transitoire, permettant de traiter la nouveauté jusqu'à ce que sa répétition la rende assimilable par le néocortex.

Enfin, les mécanismes d'attention active seraient dus aux chemins indirects de l'information passant par le thalamus, qui constitueraient un court-circuit du mécanisme de regroupement de cette information en séquences connues. Ce mécanisme peut s'activer sur commande du haut de la hiérarchie aussi bien que lors de l'arrivée d'un stimulus inconnu.

D'autres modèles de circuits neuronaux basés sur les colonnes corticales et permettant un codage prédictif et une forme d'inférence bayésienne ont été proposés dans [Bastos et al., 2012]. Ils utilisent toutefois d'autres hypothèses, comme le rôle de relais sensoriel feed-forward du thalamus et l'existence d'une population spécifique

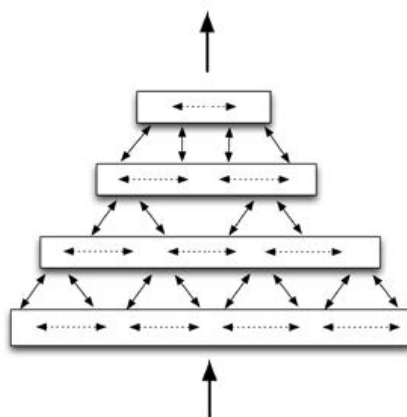


FIGURE 4.12 – Hiérarchie de régions HTM

de neurones inhibiteurs.

4.2.4.5 L'algorithme de mémoire temporelle et hiérarchique

En se basant sur ce modèle, Hawkins propose un méta-algorithme d'apprentissage automatique inspiré du fonctionnement mémoriel et prédictif, nommé HTM (Hierarchical Temporal Memory). Cet algorithme permet donc d'inférer les causes à plus ou moins haut niveau des observations qu'on lui donne comme entrées sous forme de séquences temporelles. À partir de cette modélisation complexe et hiérarchique, il peut également générer des prédictions.

Cet algorithme utilise un réseau d'unités appelées nœuds réparties en différentes régions, elles-mêmes assemblées en différents niveaux. Chaque nœud a le même fonctionnement. Il reçoit des données du niveau inférieur sous forme de séquence temporelle et calcule s'il s'active ou non. Cette information d'activation est transmise aux nœuds du niveau supérieur. Les données sensorielles entrent au niveau le plus bas. Le nœud utilise ensuite ces informations pour identifier et mémoriser des motifs spatiaux, qui sont des combinaisons des nœuds du niveau inférieur qui ont tendance à s'activer simultanément, ainsi que leur séquence temporelle. L'architecture de ce réseau est schématisée à la figure 4.12.

Comme dans le modèle mémoire-prédiction, les niveaux les plus hauts mémorisent les informations les plus abstraites en réutilisant les motifs appris par les nœuds de niveau inférieur. Leur invariance dure plus longtemps.

Un nœud vise à modéliser le fonctionnement d'une macro-colonne corticale, soit environ 80 à 100 colonnes. Cet algorithme correspond au fonctionnement des couches corticales 3, 4 et 2 (qui est supposée être fonctionnellement équivalente à la couche 4). On peut donc noter que cet algorithme ne propose pas, dans sa version actuelle, d'équivalent au comportement ou au contrôle moteur.

Implémentation et applications Hawkins a fondé l'entreprise Numenta, ensuite rebaptisée Grok Solutions, afin de développer cet algorithme et ses applications. Numenta a publié des white papers détaillant les algorithmes employés ainsi que divers logiciels, d'abord de démonstration, puis un kit de développement, puis à visée commerciale.

En tant que méta-algorithme, HTM a connu plusieurs implémentations en algorithmes. La première implémentation de HTM par Numenta a pris la forme d'un modèle hiérarchique de réseaux bayésiens [George and Hawkins, 2005]. Toute information est convertie en probabilités : ainsi, pour la reconnaissance, un nœud va estimer la probabilité de correspondance de l'entrée à chacune des catégories qu'il permet de modéliser. C'est ce jeu de probabilités qui constitue l'information transmise d'un niveau à l'autre. Il n'y a pas d'équivalent au mécanisme d'inhibition synaptique.

Une deuxième implémentation a pris une forme similaire à un réseau de neurones. Nous allons détailler son fonctionnement dans la partie suivante.

La première implémentation a donné lieu à une API intitulée NuPIC, abréviation de Numenta Platform for Intelligent Computing, écrite en C++ et en Python. À partir de cette API, ils ont développé un logiciel de démonstration, Numenta Vision Toolkit, logiciel de reconnaissance d'images, où l'utilisateur peut entrer ses propres images et catégories. Le logiciel peut proposer des topologies habituellement bien adaptées au problème de la vision.

Ils ont ensuite développé des applications à visée commerciale :

- Vitamin D Video, une application de vidéosurveillance visant à détecter la présence d'humains et d'objets en déplacement dans un flux vidéo ;
- EDSA Power Analytics, un outil permettant de surveiller un réseau électrique afin de détecter et diagnostiquer des anomalies ;
- iResemble, une application pour iPhone qui utilise le Vision Toolkit pour détecter les ressemblances entre personnes.

HTM a été utilisé avec succès pour de la reconnaissance visuelle de caractères manuscrits [Thornton et al., 2006, Thornton et al., 2008], de la classification d'images [George and Hawkins, 2005, George and Hawkins, 2009] ainsi que de chiffres lus par un voix humaine [van Doremalen and Boves, 2008].

HTM a été notamment utilisé par d'autres équipes pour classifier les mouvements humains [Hawkins et al., 2009], voire tenter de les reproduire [Seok and Kim, 2008], et classifier et reproduire des morceaux de musique [Maxwell and Eigenfeldt, 2008]. Ces tâches sont généralement considérées comme très difficiles pour des algorithmes informatiques.

4.2.4.6 L'algorithme d'apprentissage cortical

Dans le nouvel algorithme, baptisé CLA (Cortical Learning Algorithm) [Numenta, 2010], Numenta cherche à se rapprocher de la réalité biologique par rapport à leur première implémentation basée sur des réseaux bayésiens. Nous allons généralement parler de HTM-CLA pour désigner cet algorithme. Cette nouvelle version ne peut toutefois pas tout à fait être qualifiée de réseau de neurones dans la mesure où l'unité de base

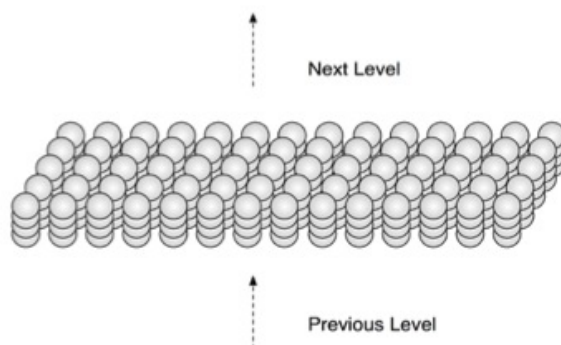


FIGURE 4.13 – Une région HTM-CLA

n'en est pas le neurone, mais la colonne.

Un réseau CLA est composé d'un ensemble de régions ordonnancées en hiérarchie. Chaque région est elle-même composée d'un tableau à deux dimensions de colonnes, dont l'activation est purement binaire. Une colonne contient plusieurs cellules. Il existe deux types de connexions : une reliant une colonne d'une région à une colonne d'une région supérieure ou inférieure dans la hiérarchie, et une reliant une cellule à une autre cellule de la même région. Le premier type de connexion est qualifié de synapse proximale, et le deuxième de synapse distale. Cette terminologie correspond à une modélisation des neurones pyramidaux, qui ont ce type de distinction entre dendrites proximales et distales. La figure 4.13 présente une région contenant des cellules agencées en colonnes.

Ces synapses sont regroupées en dendrites. Chaque colonne possède une seule dendrite proximale, mais chaque cellule possède plusieurs dendrites distales (le nombre proposé par Numenta est de une à deux douzaines). Il suffit qu'une seule dendrite distale s'active pour que toute la colonne s'active. De même, l'activation de la dendrite proximale active la colonne. En fait, la localisation de la dendrite proximale au niveau de la colonne et non de la cellule correspond à une mise en commun des dendrites proximales de toutes les cellules, qui, étant dans une colonne, ont toutes le même champ réceptif.

Chaque colonne correspond à un motif. Avoir plusieurs cellules par colonne sert à modéliser le contexte temporel de l'activation d'un motif ; il peut effectivement y en avoir plusieurs. Les connexions entre cellules correspondent au mécanisme de prédiction au sein d'une même région : dans le cadre de ce contexte temporel, je sais que ce motif est souvent suivi de cet autre ; je me prépare donc à son activation, dans ce contexte. Ainsi, une cellule peut être active, inactive ou prédire son activation.

Chaque synapse a une force de connexion, ou poids, appelée permanence. Un seuil permet de décider si la permanence de la synapse est suffisante pour que la synapse soit réellement connectée ou non.

Initialement, les connexions se font avec des colonnes et cellules choisies aléatoi-

rement, et avec un poids choisi également aléatoirement, mais de façon à être proche du seuil de connexion de la synapse.

Au sein d'une région, une inhibition locale empêche un trop grand pourcentage des colonnes d'être actives en même temps. On obtient donc une représentation clairsemée, dont nous avons vu à la partie 4.2.1 les avantages : forte représentativité, bonne résistance au bruit. Ici, le degré de clairsemage de la représentation est fixé dans l'algorithme. L'ordre de grandeur du pourcentage de colonnes actives à tout instant est proposé autour de 2%.

L'algorithme d'apprentissage et d'inférence proposé pour le réseau est constitué des trois phases suivantes :

1. Représentation spatiale : On calcule l'activation des colonnes. Une première passe calcule une quantité d'activation de chaque colonne en additionnant le nombre de ses entrées qui sont actives et en comparant ce nombre à un certain seuil pour éviter de représenter des motifs triviaux. Une deuxième passe rend la représentation clairsemée par inhibition locale. En mode apprentissage, on veille également à ce qu'aucune colonne ne soit sur- ou sous-utilisée, et on récompense les synapses proximales qui ont contribué à l'activation de la colonne, tout en pénalisant les autres.
2. Mise en contexte temporel : On calcule l'activation des cellules des colonnes actives. Si aucune cellule ne prédisait son activation, alors toutes les cellules sont activées : tous les contextes temporels sont en effet possibles. Sinon, seules les cellules qui prédisaient leur activation sont activées, puisque leur contexte temporel est probablement bon. En mode apprentissage, on récompense également les prédictions correctes d'activation, comme décrit à l'étape suivante. Si au contraire la cellule est passée d'active à inactive, on annule toute modification de synapses prévue au titre d'une prédiction qui ne s'est pas réalisée.
3. Prédiction : On calcule la prédiction des cellules. Si suffisamment de cellules connectées à une même dendrite sont actives, alors cette dendrite est active et sa cellule prédit. En mode apprentissage, on prévoit de récompenser ou pénaliser les synapses distales qui ont contribué ou non à l'activation de la dendrite, mais on reporte cette modification, qui sera déclenchée plus tard à l'étape de mise en contexte temporel, pour ne le faire que sur les prédictions correctes. Au cas où aucune synapse n'aurait permis la prédiction qui aurait manqué, on va alors chercher à créer de nouvelles synapses et de nouvelles dendrites distales.

La figure 4.14 présente une région où certaines cellules sont actives (gris clair), certaines cellules prédisent leur activation (gris foncé) et où, en plusieurs endroits, l'échec de la prédiction implique l'activation de toutes les cellules d'une même colonne.

Comme on peut le remarquer, on n'a décrit ici qu'un seul sens de l'algorithme HTM : les connexions montantes. Sans le feed-back des régions supérieures, l'algorithme est incomplètement spécifié, ce qui est certainement volontaire de la part de Numenta, qui souhaite commercialiser son implémentation de cet algorithme. D'autre

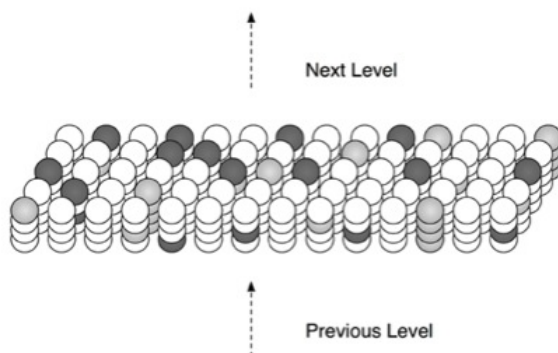


FIGURE 4.14 – Prédiction au sein d'une région HTM-CLA

part, cet algorithme correspond uniquement au fonctionnement de la couche 4 du néocortex⁵.

On voit que ce modèle comporte des éléments bien plus proches de la biologie du néocortex que les réseaux de neurones artificiels plus classiques.

4.2.4.7 Autres modèles similaires

L'algorithme HTM-CLA combine et élargit des aspects des réseaux bayésiens, du partitionnement temporel et spatial et de l'apprentissage profond, tout en utilisant un réseau d'unités semblable à un réseau de neurones. Nous avons expliqué dans les parties précédentes à quoi correspondaient ces notions, et nous venons maintenant d'expliquer en détail le fonctionnement de HTM-CLA. Nous pouvons donc maintenant mieux analyser les ressemblances et les différences entre ces modèles.

Réseaux bayésiens Comme les réseaux bayésiens, les HTM sont constitués d'un ensemble de nœuds en hiérarchie arborescente, et chaque nœud dans la hiérarchie a pour fonction de découvrir les causes sous-jacentes à ses entrées. Le transfert d'information dans les deux sens, feed-forward et feed-back, est également commun aux deux modèles. D'ailleurs, la première implémentation de HTM choisissait d'utiliser des réseaux bayésiens. Par ailleurs, les chercheurs Tai Sing Lee et David Mumford ont proposé une théorie du fonctionnement du cortex basée sur de la propagation de croyance bayésienne et hiérarchique [Lee and Mumford, 2003], qui ne traite toutefois pas des représentations invariantes, mais se base sur les modèles génératifs et le codage prédictif.

En outre, il existe un autre modèle d'apprentissage bayésien, DeSTIN [Arel et al., 2009], qui met l'accent sur l'aspect spatio-temporel des données.

Cependant, ces deux modèles restent assez différents. En effet, les HTM concernent des flux de données évoluant dans le temps, alors que les réseaux bayésiens n'ont

5. Cela a été précisé par Jeff Hawkins : « The CLA specifically deals with what a layer of cells does. »

pas cette forte dimension temporelle. En outre, si la structure des réseaux bayésiens doit être fixée à l'avance, ou alors apprise par d'autres méthodes d'apprentissage automatique, et que les concepts modélisés doivent absolument être fournis, les concepts représentés par les HTM et les liens entre eux sont découverts par ce même algorithme. Leur usage peut donc être assez différent. Enfin, bon nombre de modèles bayésiens utilisent des algorithmes non locaux, donc peu bio-réalistes.

Réseaux de neurones Puisqu'ils tentent tous deux de modéliser le fonctionnement du néocortex et des neurones, on peut facilement faire un lien entre les HTM et les réseaux de neurones. Cependant, les unités employées dans les HTM ne sont pas fonctionnellement équivalentes à un seul neurone, mais à une micro-colonne corticale (80 à 100 neurones). La modélisation est donc faite à un niveau d'abstraction légèrement supérieur : on cherche à reproduire le fonctionnement d'une unité de base légèrement plus abstraite, mais l'importance est mise sur le fait que cette unité serait en fait l'unité de base algorithmique du néocortex. Ainsi, si l'on pourrait reproduire ce fonctionnement avec une modélisation employant des neurones, cela ne serait pas nécessairement le cas, car les connexions entre ces neurones et les modèles de neurones employés pourraient être très différents, et donc ne pas correspondre au fonctionnement du néocortex. D'autres part, les topologies traditionnellement employées dans les réseaux de neurones ne correspondent pas nécessairement à un fonctionnement hiérarchique.

HTM-CLA fait également la distinction entre dendrites proximales et distales, alors que les connexions dans les réseaux de neurones classiques correspondent uniquement aux dendrites proximales. Il propose également une modélisation plus poussée du rôle de l'inhibition synaptique. Enfin, le rôle des synapses y est assez différent, puisque l'apprentissage dans les réseaux de neurones classique consiste principalement en une modification du poids synaptique, dont le réglage doit être très fin, alors que HTM-CLA propose des modifications bien plus grossières, mais propose également la création de nouvelles synapses.

Les modèles de réseaux de neurones les plus proches des HTM que nous avons vus sont les modèles ART (présentés en annexe D) et les modèles de réseaux de neurones à convolution, qui relèvent également de l'apprentissage profond. Dans ART, on retrouve l'idée que la reconnaissance et la classification au sein d'un réseau de neurones résulte de l'interaction entre les messages venus du haut, qui correspondent aux attentes et prédictions, et ceux venus du bas, qui transmettent les informations sensorielles. Cependant cette idée est réalisée différemment, en deux modules distincts, configuration qui nous semble moins bio-réaliste et n'apporte pas de fonctionnalités particulières à part une taille minimale.

Apprentissage profond Les modèles d'apprentissage profond sont également proches des HTM par leur structuration en hiérarchie, et correspondent également à une volonté de modéliser le néocortex. Une partie de ces modèles emploie d'ailleurs des réseaux de neurones.

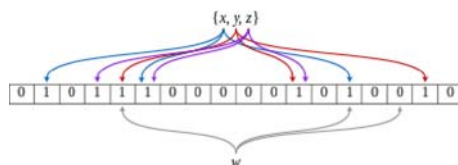


FIGURE 4.15 – Représentation d'un filtre de Bloom

Filtre de Bloom Le modèle de mémoire proposé par un système basé à la fois sur une représentation clairsemée de la mémoire et sur un échantillonnage des composants mémorisés se rapproche du filtre de Bloom. Il s'agit d'une structure de données probabiliste compacte, qui fait appel à une fonction de hachage et permet de dire avec certitude qu'un élément n'est pas dans l'ensemble, mais ne permet que d'affirmer qu'il y est probablement. La probabilité de faux positif est de

$$\left(1 - \left[1 - \frac{1}{m}\right]^{nk}\right)^k \approx \left(1 - e^{-\frac{kn}{m}}\right)^k$$

avec m le nombre de bits de l'ensemble, n le nombre d'éléments stockés et k le nombre de fonctions de hachage. La figure 4.15 présente une représentation d'un filtre de Bloom.

Toutefois, les filtres de Bloom ne cherchent aucunement à modéliser l'aspect temporel dont nous avons montré qu'il est particulièrement important à la formation de la mémoire et à la prédiction.

Conclusion

L'étude du fonctionnement du cerveau humain présente non seulement un intérêt manifeste au vu des performances humaines en termes d'apprentissage, mais donne également lieu à divers algorithmes et méthodes d'apprentissage automatique, dont certains sont très performants. Nous avons vu deux grands axes de bio-inspiration : les réseaux de neurones, cherchant à reproduire le fonctionnement connexionniste des neurones et synapses à divers degrés de réalisme, et l'apprentissage profond, qui insiste sur une architecture hiérarchique de diverses régions afin d'apprendre plus efficacement des causes complexes et abstraites.

Bien entendu, il y a une forte intersection entre ces deux axes, et c'est là que nous pensons donc trouver les méthodes les plus prometteuses. Notamment, HTM-CLA nous a paru sortir du lot par le paradigme mémoire-prédiction qui l'a inspiré et qui explique particulièrement bien le fonctionnement du néocortex humain, par son usage d'un apprentissage non supervisé et incrémental, ainsi que par ses capacités de prédiction dans le temps, qui nous semblent un bonus très intéressant. L'approche HTM-CLA correspond notamment au compromis considéré comme le plus prometteur par Demis Hassabis (mentionné lors du Singularity Talk 2010), qui combine l'approche de l'intelligence artificielle avec celle des neurosciences, en visant une

compréhension fonctionnelle plutôt qu'une reproduction à grande échelle du cerveau dans ses moindres détails comme le font les projets Blue Brain [Markram, 2006] et Synapse (DARPA) [Ananthanarayanan and Modha, 2007].

4.3 Proposition d'utilisation de HTM-CLA pour le suivi des étudiants

Introduction

Dans la partie précédente, nous avons choisi d'étudier plus en détail les algorithmes d'apprentissage automatique inspirés du fonctionnement du cerveau humain, dont l'étude présente un grand intérêt au vu de ses performances, et plus spécifiquement les algorithmes connexionnistes, qui ont le double avantage d'être plus biologiquement plausible et d'être mieux adaptées à l'apprentissage et à la généralisation de connaissances.

Après avoir présenté les connaissances biologiques actuelles sur le fonctionnement du cerveau, nous avons recensé les différents algorithmes qui relèvent de ces critères, en portant une attention particulière aux algorithmes d'apprentissage non supervisés et incrémentaux, qui sont bien moins contraignants d'utilisation et qui correspondent également davantage à l'apprentissage humain, lui-même non supervisé et incrémental.

Nous avons vu deux grandes familles d'algorithmes, avec un recoupement entre elles, qui sont les réseaux de neurones et l'apprentissage profond. Nous avons également étudié la question du codage neuronal et vu qu'un codage clairsemé semblait offrir les meilleurs avantages. À l'intersection de ces deux familles se trouvent plusieurs algorithmes, mais un en particulier, HTM-CLA⁶, nous a semblé sortir du lot par son paradigme clair et explicatif qui vise une compréhension fonctionnelle plutôt qu'une reproduction à grande échelle du cerveau dans ses moindres détails, par sa correspondance à nos divers critères (apprentissage incrémental, non supervisé, codage clairsemé) et par ses possibilités, notamment de prédiction.

En outre, nous avons également montré que les hypothèses visant à le rendre plus performant sont bien adaptées à notre problème. Il nous semble ainsi justifié de l'utiliser dans le cadre du suivi des élèves apprenant en ligne. Nous allons maintenant détailler les adaptations nécessaires à cette utilisation.

4.3.1 Entrées et sorties

Un réseau HTM prend comme entrée un ensemble de données qui évolue dans le temps, sous la forme d'un flux de ces données. Il n'est nulle part précisé dans la description de l'algorithme si ce flux doit être synchrone ou asynchrone. Il nous semble plus pertinent d'envisager une utilisation asynchrone pour notre application, dans la mesure où les activités en ligne des étudiants ne présentent pas de régularité

6. Pour rappel : ce sigle signifie la version Cortical Learning Algorithm du méta-algorithme Hierarchical Temporal Memory.

dans le temps. Les logs sur lesquels nous allons nous baser ne sont pas prélevés à intervalles réguliers mais bien de façon asynchrone.

Les données asynchrones qui serviront d'entrée seront les données que nous avons listées dans la partie 2.1. Elles recouvriront donc les traces d'utilisation et les données extérieures. C'est donc parmi ces données que l'algorithme va essayer de repérer des motifs spatio-temporels.

Nous devons convertir ces données en un format compréhensible par un réseau HTM-CLA. Pour cela, nous devons disposer de quelque chose qui ressemble à une région, en comportant un ensemble d'éléments binaires (que la région située immédiatement au-dessus perçoive comme des colonnes), et qui respecte la contrainte de codage clairsemé. L'avantage de cette conversion est qu'il est très facile de fournir une valeur seulement à certaines entrées et pas à d'autres (par exemple, quand on entre une trace correspondant à la lecture d'une leçon, il n'y a pas de valeur à la variable « note obtenue », ce qu'il faut modéliser sans qu'il y ait ambiguïté avec le fait d'obtenir une note de 0), en n'activant tout simplement aucun des éléments correspondant à ces variables non pertinentes.

Nous récapitulons la situation de nos entrées dans le schéma 4.16.

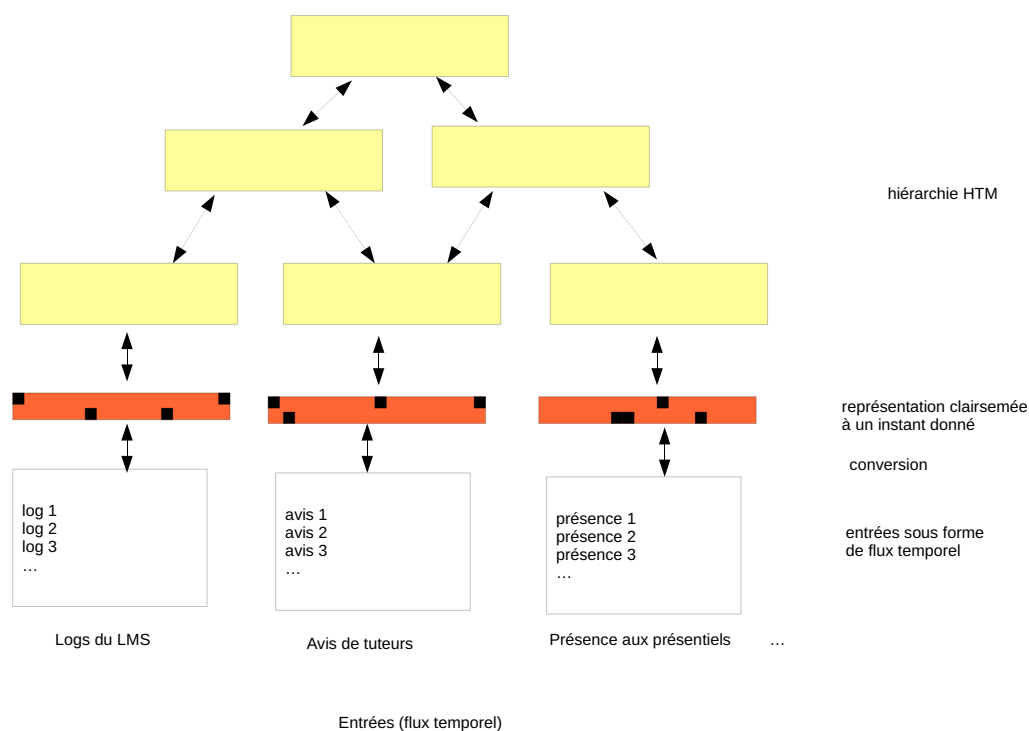


FIGURE 4.16 – Entrées de HTM

En sortie, nous souhaitons pouvoir utiliser au maximum les diverses possibilités

offertes par l'apprentissage automatique et par l'algorithme HTM-CLA. Il est nativement conçu pour l'inférence, l'explication, la prédiction et la simulation, que nous souhaiterons donc réaliser.

L'inférence consiste à estimer certaines variables à partir de variables observables. Avec l'inférence, nous allons donc chercher à connaître la valeur de certaines variables estimées comme des causes sous-jacentes, et si possible des liens entre elles. Ainsi, nous pourrions voir si le réseau construit des concepts correspondant aux notions de jour, matin, après-midi, soir, nuit, week-end, semaine, vacances scolaires ; aux notions de cours, des différents concepts maîtrisés ou non par les élèves, la quantité de travail fournie par un élève, la motivation d'un élève, etc. Il serait également intéressant de voir si notre IGA se retrouve dans le réseau.

L'explication est le phénomène inverse de l'inférence : on observe le réseau créé et on cherche à en comprendre le sens en le traduisant en variables compréhensibles par un humain. Avec l'explication, nous souhaitons obtenir une liste de variables nouvelles correspondant aux motifs trouvés dans les données ainsi qu'aux causes fondamentales des comportements des élèves.

Avec la prédiction, nous obtiendrons des prédictions à différents niveaux. Au niveau le plus bas, celui des entrées, nous obtiendrons les prochaines actions des élèves. Aux niveaux plus élevés, nous obtiendrons des prédictions plus abstraites pour les variables cachées qui auront été découvertes.

Une simulation correspond à une prédiction sur la base de données fictives. Avec une simulation, on pourrait par exemple réussir à convaincre un élève que s'il passe 10 minutes de plus par jour sur la plate-forme, ses résultats s'amélioreront significativement.

Si possible, sous réserve d'arriver à adapter l'algorithme, nous souhaitons également l'utiliser à des fins de partitionnement de données, de classification, de régression et de détection d'anomalie.

Avec le partitionnement de données, nous chercherons à créer des groupes d'élèves de comportement similaire, pour comprendre ensuite les raisons de ce regroupement, et les profils de nos élèves. Nous pourrions aussi chercher à regrouper les activités de même difficulté, ou ayant trait aux mêmes notions, afin de créer des regroupements pertinents pour analyser plus finement les autres résultats.

Avec la classification, nous chercherons principalement à distinguer les élèves ayant réussi leur formation de ceux y ayant échoué. Armés de cela, nous chercherons à expliquer le modèle obtenu en termes des facteurs influant sur la réussite d'un élève, et en l'appliquant à des élèves en cours de formation, à prédire leur succès, leur échec ou leur abandon futur.

Avec la régression, nous chercherons à créer un modèle des différentes notes obtenues par les élèves, aussi bien aux activités notées qu'aux examens finaux. Comme pour la classification, nous nous en servirons ensuite à la fois pour comprendre le modèle et à le réutiliser pour prédire des résultats futurs.

Avec la détection d'anomalie, on pourra détecter si un élève se connecte anormalement peu souvent ou s'il a une note anormalement basse et faire remonter cette information au responsable de formation pour qu'il juge de la conduite à tenir. Nous

pourrions également comparer le parcours suivi par l'élève à travers les ressources à divers parcours, pré-définis ou vus sur les instances précédentes de la formation, et détecter s'il suit un parcours semblant étrange ou incohérent.

Ces différentes propositions de sorties sont résumées à la figure 4.17.

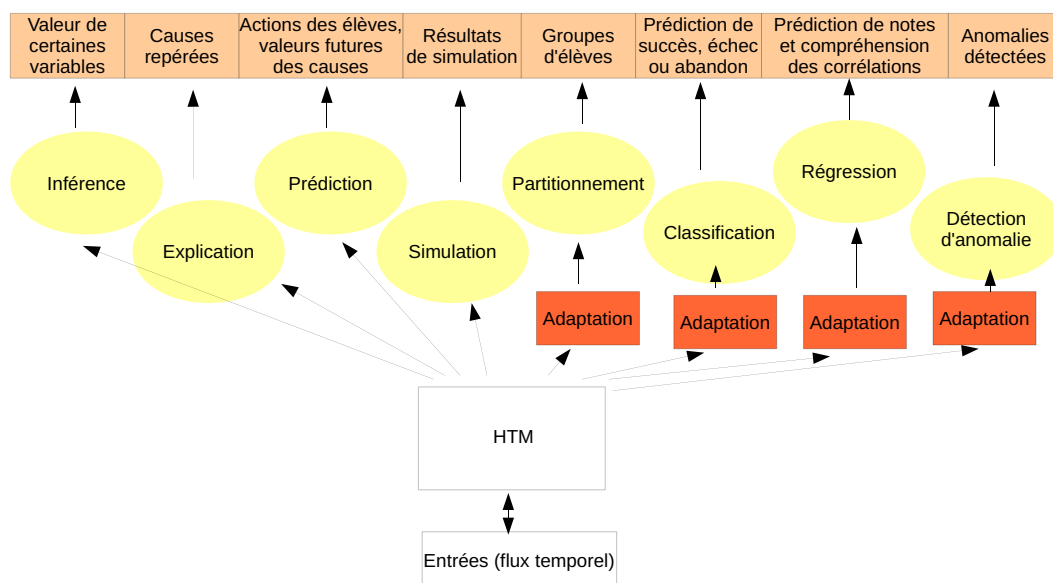


FIGURE 4.17 – Sorties de HTM

Tout cela contribue à définir plusieurs modèles intéressants : un modèle causal des différentes variables influant sur une formation en ligne, une liste de profils d'élèves, un outil capable de prédire le succès ou l'échec d'un élève, avec un niveau de détail allant jusqu'à ses notes, et un outil de simulation unique en son genre.

4.3.2 Adaptation de HTM-CLA à nos sorties

L'algorithme HTM-CLA est nativement capable de deux fonctionnalités : l'inférence et la prédiction. Cependant, nous pensons pouvoir l'adapter afin de fournir des sorties d'un autre type. Cela nous permettra d'effectuer des comparaisons entre les résultats obtenus par ces méthodes et les résultats obtenus par les méthodes plus classiques. Nous nous attendons à une plus grande pertinence des résultats obtenus, mais à un temps de calcul bien plus important.⁷

7. Ceci est confirmé par Subutai Ahmad, le vice-président de la division ingénierie de Numenta : « Les HTM ne sont pas plus efficaces en termes de temps de calcul. Il s'y passe bien plus de choses.

Nous allons ici expliquer comment nous avons réalisé ces adaptations. Pour cela, nous allons avoir besoin de rentrer un peu dans les détails de cet algorithme.

Pour rappel, un réseau CLA est composé d'un ensemble de régions ordonnancées en hiérarchie. Chaque région est elle-même composée d'un tableau à deux dimensions de colonnes, dont l'activation est binaire et qui utilise un codage clairsemé. Une colonne contient plusieurs cellules. Chaque colonne possède une seule dendrite proximale qui la relie aux colonnes de la région inférieure, mais chaque cellule possède plusieurs dendrites distales qui la relie aux cellules de cette même région.

Chaque colonne correspond à un motif. Avoir plusieurs cellules par colonne sert à modéliser le contexte temporel de l'activation d'un motif ; il peut effectivement y en avoir plusieurs. Les connexions entre cellules correspondent au mécanisme de prédiction au sein d'une même région dans le cadre de ce contexte temporel.

4.3.2.1 Inférence

Nous avons mentionné que l'inférence consiste à estimer certaines variables à partir de variables observables. Ici, les variables observables sont nos données d'entrée. Dans le réseau CLA vont se trouver représentés un certain nombre de motifs, dont on pourrait souhaiter connaître l'activation ou non. Toute la difficulté ici va donc consister à faire le lien entre un motif donné sous la forme d'un ensemble de colonnes et sa représentation compréhensible par un humain.

- Pour ce faire, on peut utiliser deux méthodes, qu'on peut également combiner :
- connaître la valeur à certains moments de la variable qu'on souhaite estimer, et la comparer aux motifs actifs à ce moment ;
 - connaître les motifs correspondant à certains composants observables de cette variable (pour les régions de niveau inférieur) ou à certaines causes possibles observables de cette variable (pour les régions de niveau supérieur) et observer quelles colonnes ont des liens avec eux.

4.3.2.2 Explication

Les méthodes utilisées sont toutefois très proches de celles utilisées pour l'inférence, puisqu'on va chercher à se ramener à des motifs déjà compris et à des variables observables.

4.3.2.3 Prédiction

La prédiction est un des mécanismes de base de HTM-CLA. Il faut là encore pouvoir traduire ces prédictions cellulaires en prédictions compréhensibles par un humain.

Au niveau élémentaire, celui des entrées fournies au réseau, on dispose de prédictions directement compréhensibles sur les variables fournies. Pour toute autre

L'algorithme a une vitesse raisonnable, il met environ 25 ms sur mon ordinateur portable pour un point de données. Les modèles à moyenne flottante sont probablement plus rapides d'un ordre de grandeur. » Source : mailing list NuPIC-discuss, traduction personnelle.

variable et tout autre niveau, il faut passer par les processus d'inférence, pour savoir identifier le lieu de la prédiction qui nous intéresse, ou d'explication, pour comprendre quelles prédictions sont effectivement réalisées par le réseau.

4.3.2.4 Simulation

La réalisation d'une simulation avec HTM-CLA est extrêmement simple, puisqu'il suffit de désactiver temporairement le mode d'apprentissage du réseau, de présenter les entrées correspondant aux hypothèses de la simulation, puis d'observer la prédiction fournie par le réseau.

4.3.2.5 Partitionnement

Nous arrivons ici dans les mécanismes non nativement inclus dans HTM-CLA. Nous avons toutefois cherché à proposer un mécanisme de partitionnement afin d'en tester les résultats et de les comparer aux méthodes de partitionnement classiques.

En premier lieu, il faut décider comment traiter la composante temporelle forte inhérente aux HTM. Deux options s'offrent à nous : le partitionnement à un instant donné, ou le partitionnement moyen sur une période. Dans le premier cas, nous allons observer le réseau à un seul pas de temps, et dans le deuxième, nous allons l'observer sur ladite période. Si l'on souhaite partitionner uniquement un nombre d'instances spécifique, nous allons présenter successivement ces instances et réaliser nos observations sur la période de ces présentations.

Ensuite, nous allons étudier les motifs générés par les entrées sur la période de temps choisie dans une région donnée. Nous allons choisir la région la plus haute de la hiérarchie, qui est la plus abstraite et nous semble la plus à même de fournir les explications sous-jacentes à un partitionnement pertinent.

Dans cette région, nous allons comparer les motifs générés par les différentes instances correspondant aux différents pas de temps. Pour effectuer cette comparaison, nous allons nous baser sur le nombre de colonnes communes. Cette comparaison est pertinente du fait du codage clairsemé et ne le serait pas sinon : les colonnes communes correspondent assez probablement à un élément sémantique commun entre les motifs.

À partir de cette comparaison, diverses possibilités s'offrent à nous. Soit on souhaite fixer un nombre de groupes donné, auquel cas nous allons fusionner les motifs les plus proches jusqu'à en obtenir le nombre désiré, de façon comparable à un partitionnement hiérarchique ascendant. Sinon, on peut décider d'un seuil de proximité à partir duquel ce regroupement est fait de façon automatique. On obtient ainsi un certain nombre de méta-motifs correspondant à un ensemble de colonnes plus grand que le nombre normalement activé dans la région (du fait du codage clairsemé).

On peut ensuite effectuer une projection de ce partitionnement sur une variable donnée. Dans notre exemple, si on cherche à faire des groupes d'élèves, la variable sur laquelle sera faite la projection sera l'identifiant de l'élève. Pour ce faire, nous

allons faire appel à la phase descendante de l'algorithme d'apprentissage, qui va faire redescendre l'équivalent du motif jusqu'à la couche contenant la variable en question (généralement la couche d'entrée) et en activer les composantes recherchées. Si les entrées présentées étaient une liste d'instances contenant cette variable, on peut également et plus simplement associer la valeur de la variable de projection à chaque motif repéré, et les regrouper lors de la fusion de motifs. Nous allons alors obtenir un groupe d'instances de la variable souhaitée, associées à un méta-motif.

Il pourrait être possible d'étendre cette proposition à un modèle étudiant plusieurs régions simultanément, mais nous avons choisi de ne pas le détailler pour en rester à une proposition simple.

4.3.2.6 Classification

La classification se base sur le principe proposé pour notre méthode de partitionnement. En effet, on associe aux motifs repérés durant l'apprentissage une étiquette correspondant à la classe à laquelle l'instance appartient, et on regroupe les motifs suivant cette étiquette. Enfin, on fait passer un jeu de données de test, on observe les motifs activés et on renvoie la classe correspondant à l'étiquette du méta-motif le plus proche.

4.3.2.7 Régression

Nous allons nous intéresser au cas le plus simple, celui de la régression linéaire, et chercher à en estimer les coefficients. Pour ce faire, nous allons successivement lancer des simulations dans lesquelles toutes les variables scalaires sauf une prennent la valeur nulle. Nous obtenons ainsi une estimation directe du coefficient associé à la variable non nulle. Évidemment, cette estimation est biaisée par les valeurs totalement inhabituelles des autres variables. Toutefois, cette méthode a le mérite d'être très simple et de fournir un premier résultat, à raffiner ensuite.

4.3.2.8 Détection d'anomalie

La détection d'anomalie est assez proche d'être une partie native de l'algorithme, mais elle ne figure toutefois pas dans la description de l'algorithme CLA. Elle était toutefois décrite dans le livre *On Intelligence* [Hawkins and Blakeslee, 2004] sous la forme de la prédiction erronée, qui provoque une prise de conscience et une remontée dans la hiérarchie. La forme que prend cette remontée n'était toutefois décrite qu'en termes de neurones, et proposait un rôle de la couche 3 du néocortex servant à différencier les séquences prédites des séquences non prédites. Il reste toutefois très flou sur le traitement qui en est fait dans la région supérieure.

En termes de CLA, l'équivalence à cette couche 3 se trouve dans les différents états d'activation possibles des cellules, qui ne sont toutefois pas remontés au niveau des colonnes. Or, dans leur modèle, la communication entre régions se fait uniquement au niveau des colonnes. Nous allons donc devoir nuancer l'état d'activation des colonnes

pour qu'il comporte également cette information afin qu'elle puisse remonter au niveau supérieur.

Dans un document plus récent centré sur la détection d'anomalie [Numenta, 2013], Numenta décrit sa façon d'implémenter la détection d'anomalie en utilisant CLA. Pour ce faire, ils calculent un score d'erreur de prédiction sur toute la région (ou toutes les régions). Une moyenne flottante sur un intervalle de temps court de ce score (environ une heure) est comparée à sa moyenne flottante sur une durée plus longue (quelques semaines). Cette comparaison sert de base au calcul d'un score d'anomalie.

En l'absence d'autre précision apportée, on peut supposer qu'aucune autre utilisation n'est faite de l'erreur de prédiction. Effectivement, ce mécanisme est suffisant pour faire une détection d'anomalie. Toutefois, nous aimerions aller un peu plus loin et permettre de réexaminer les connaissances à la lumière de cette information. Selon la description faite dans [Hawkins and Blakeslee, 2004], le traitement doit être dans un premier temps de proposer une nouvelle séquence plus adaptée, et si cela échoue, de propager l'information plus haut dans la hiérarchie.

4.3.3 Implémentation

4.3.3.1 Implémentation du cœur HTM

Au début de ce projet, nous avons cherché des bibliothèques Java reprenant les spécifications de l'algorithme HTM-CLA, et n'en avons pas trouvé de satisfaisante. L'utilisation du langage Java était une contrainte due à la future intégration avec les applications d'Andil. À ce moment, les bibliothèques proposées par Numenta utilisaient la première version de l'algorithme HTM, qui était essentiellement bayésienne, sous l'influence de Dileep George. Les autres bibliothèques trouvées n'utilisaient pas les bons langages ou ne nous paraissaient pas respecter suffisamment bien les spécifications. Nous avons donc choisi de réaliser notre propre implémentation de HTM-CLA, en se fondant sur le livre *On Intelligence* [Hawkins and Blakeslee, 2004] et sur le rapport technique [Numenta, 2010] comme spécifications.

Nous présentons en annexe B des détails liés à la conception de notre implémentation. Dans la figure B.1 figure le diagramme de classes auquel nous a mené notre conception, et un deuxième diagramme présenté en figure B.2 détaille les méthodes disponibles dans chaque classe, afin de mieux expliquer le fonctionnement de l'ensemble. Les explications associées à ces deux diagrammes de classes sont également en annexe B.

Dans les divers choix d'implémentation, nous avons fait le choix de représenter les régions comme un tableau à deux dimensions de colonnes. Nous avons également choisi de respecter la méthode proposée d'inhibition locale des colonnes, pour son bio-réalisme.

Comme nous l'avions précisé lors de la partie précédente, seule la phase montante de l'algorithme a réellement été décrite. Nous avons donc dû proposer notre propre mécanisme pour la phase descendante, qui permet d'éclaircir les entrées ambiguës et

de guider la prédiction. Le mécanisme que nous avons proposé est le suivant :

```
Hierarchy.simulate() :
for(int level = 0; level < nbLevels; level++) {
    for(Region r : regions.get(level)) {
        r.passUpInputs(learning, temporal);
    }
}
for(int level = nbLevels-1 ; level >=0 ; level--) {
    for(Region r : regions.get(level)) {
        r.passDownPrediction);
    }
}
```

Chaque région transmet ensuite cette commande à chacune de ses colonnes, qui la transmet à son segment dendritique proximal uniquement si au moins une des cellules prévoit son activation prochaine. Le segment la transmet alors à toutes ses synapses proximales, qui font descendre l'information à leur colonne d'origine, activant alors la réception de prédiction :

```
Column.receivePrediction() :
this.predictionReceived[0]++;
for(Cell c: this.cells) {
    c.receivePrediction();
}
```

```
Cell.receivePrediction() :
this.predictionReceived[0] ++;
```

Une fois l'information ainsi reçue et stockée au niveau des colonnes et des cellules, la valeur va être utilisée pour modifier le calcul de l'activation de la colonne ou de la cellule.

Le calcul de l'activité booléenne d'un segment dendritique distal donné dds, au sein d'une cellule, devient donc le suivant :

```
dds.activity(time) + this.predictionReceived[1]
>= Parameters.distalActivationThreshold
```

Et le calcul de l'activation d'une colonne devient le suivant :

```
float activity = proximal.activation()
+ Parameters.predictionWeight * this.predictionReceived[1];
if(activity < Parameters.proximalActivationThreshold) {
    activity = 0;
}
else {
```

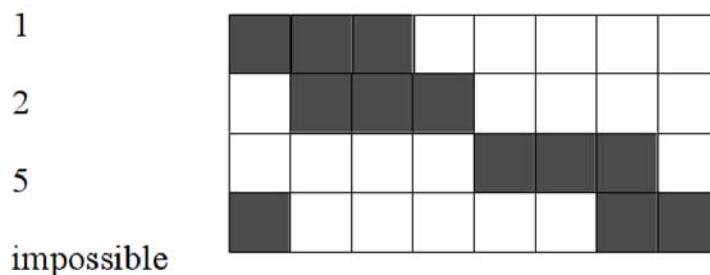


FIGURE 4.18 – Un exemple de convertisseur scalaire pour des entiers allant de 1 à 6

```

    activity *= this.boost;
}

```

Le compteur du nombre de prédictions reçues est remis à zéro à chaque nouveau pas de temps, et l'ancienne valeur est stockée dans un historique puisque le calcul qui la nécessite se fait au pas de temps suivant.

Pour la classe de conversion de nombres en représentation clairsemée, `ScalarConverter`, nous avons procédé de la façon suivante. Trois valeurs nous sont nécessaires pour initialiser notre convertisseur : un minimum et un maximum que les données numériques ont le droit d'atteindre, et une résolution.

La figure 4.18 présente notre idée pour convertir les nombres tout en maintenant un sens de leur ordonnancement et de leur continuité. Nous allons pour cela utiliser plusieurs bits contigus simultanément, qui serviront à la création du contexte sémantique des nombres qui ne pourrait sinon pas être recrée sans un long apprentissage dans une région ad hoc.

Pour cela, nous commençons par calculer le nombre de cases qui seront employées à la représentation. Nous avons choisi arbitrairement un taux de clairsemage de 5%, ce qui signifie qu'une case sur 20 doit être activée. Pour que cette valeur tombe juste, nous utilisons le plus petit commun multiple.

$$nbCases = ppcm(nbValues, 20) + 2$$

avec $nbValues = \lceil (max - min) / res \rceil$

Nous pouvons maintenant définir la fonction de conversion comme suit :

```

boolean[] convert(nombre):
    conversion = nouveau tableau de booléens initialisés à faux
    scalarValueIndex = floor((nombre-min)/res)
    scalarIndex = scalarValueIndex * nbCases/nbValues
    nextIndex = (scalarValueIndex+1)*nbCases/nbValues
    pour i de scalarIndex à nextIndex:
        conversion[i] = vrai
    retourner conversion

```

On peut de la même façon définir une fonction de conversion inverse, qui sert à comprendre quelle valeur est prédite à un moment donné dans la région d'entrée à partir de l'activation des unités binaires :

```
nombre backConvert(boolean[] cases):
  pour toutes les cases du tableau de gauche à droite:
    si la case i est active:
      retourner (i+1)*nbValues/nbCases*res+min
```

Nous avons également dû proposer des valeurs par défaut pour les différents paramètres. En nous inspirant des dimensions caractéristiques de nos problèmes, et des ouvrages cités plus haut, nous avons choisi ces valeurs :

- nombre de cellules par colonne : 4
- taux de clairsemage au sein d'une région : 0,05
- rayon d'inhibition des colonnes entre elles : 5
- nombre d'états mémorisés pour les moyennes temporelles glissantes : 100
- poids accordé à la prédiction venue de la région supérieur : 0,5
- nombre de synapses proximales dans un segment dendritique proximal : 1
- seuil d'activation du segment dendritique proximal, en nombre de synapses proximales : 1
- nombre de segments dendritiques distaux par cellule : 3
- nombre de synapses distales par segment dendritique distal : 5
- seuil d'activation du segment dendritique distal, en nombre de synapses distales : 1
- seuil de connexion d'une synapse : 0,2
- seuil d'incrémentatation de la permanence des synapses : 0,05

Pour le taux de clairsemage, nous avons jugé que sa valeur exacte n'est pas importante. Toutefois, l'ordre de grandeur nous semble correct.⁸

Numenta n'a pas encore travaillé sur le timing, qui est géré dans leur modèle par un circuit vers l'hypothalamus avec retard. On peut également imaginer qu'un long apprentissage soit nécessaire pour savoir comment convertir cette information de retard en évaluation du temps total qui passe. Nous n'avons donc pas cherché à implémenter non plus cette fonctionnalité, qui nous paraissait nécessiter une taille de réseau et un temps d'apprentissage importants.

Toutefois, l'absence de cette fonctionnalité pourrait nous manquer : il serait appréciable que le réseau puisse se créer des règles du type « dans 2h, je considérerai qu'il est suffisamment en retard pour que cela soit anormal, et je lèverai une alerte en conséquence ».

8. Nous avons trouvé sur le blog de Numenta (source : <http://numenta.com/technology/blog/the-neuroscience-behind-grok-part-2.html>) cette phrase (traduction personnelle) : « Les représentations de Grok ont généralement 2000 bits, dont seulement 2% (soit environ 40 bits) sont des 1. Ceci émule le clairsemage du cerveau, où seules quelques cellules sont actives à un moment donné. »

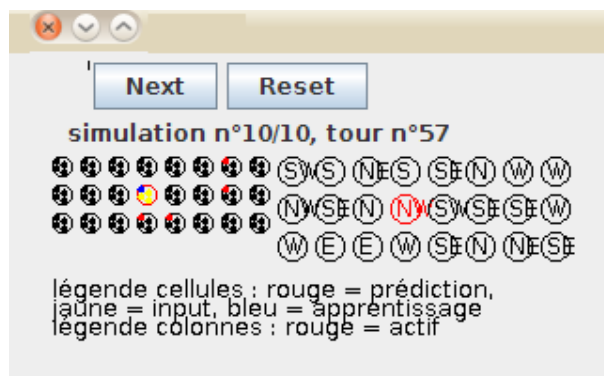


FIGURE 4.19 – Notre interface de visualisation et de contrôle de simulation HTM

4.3.3.2 Implémentation de l'apprentissage automatique

Comme expliqué plus haut, nous avons choisi de recréer diverses méthodes d'apprentissage automatique, toutes basées sur l'algorithme HTM-CLA. Il en a résulté un paquet indépendant, qu'on peut qualifier de librairie, que nous avons séparés en plusieurs sous-paquets. Le diagramme de classe correspondant est présenté à la figure B.3. Un second diagramme plus détaillé est présenté à la figure B.4.

4.3.4 Analyse des résultats obtenus

Nous avons développé plusieurs outils pour analyser le fonctionnement de HTM.

Le premier est une interface graphique rudimentaire permettant d'afficher la hiérarchie et de faire avancer la simulation pas par pas. Les différents états d'activation possibles de chaque cellule (activé, prédisant, apprenant ou inactif) sont représentés par un code couleur (rouge, jaune, bleu ou noir), et il en va de même pour les colonnes qui sont rouges si elles sont actives et noires sinon. Notre interface est présentée dans la figure 4.19.

Cette interface permet de se rendre compte au plus près de ce qui se passe dans la simulation. On peut vérifier si le nombre de colonnes activées respecte bien le critère de clairsemage de la représentation, et faire du débogage basique, comme vérifier qu'une colonne activée a bien au moins une cellule activée, et qu'aucune cellule n'est active en dehors d'une colonne active.

Cette interface graphique est codée en utilisant la bibliothèque graphique Swing nativement fournie par Java.

Un deuxième outil a été développé pour permettre d'analyser les simulations de type partitionnement de données. Lorsqu'elles ont lieu, nous produisons en sortie plusieurs fichiers au format CSV (comma-separated value) qui retracent la simulation à différents niveaux de granularité et de détail.

Un premier fichier contient la liste des colonnes actives dans la plus haute région de la hiérarchie à chaque pas de temps. On se souvient que c'est sur cet ensemble qu'est basé notre adaptation pour le partitionnement de données ; il est donc pertinent

de le suivre. Un deuxième fichier contient un récapitulatif de la simulation en termes de groupes observés, de groupes conservés si l'on a apporté une contrainte de nombre maximal de groupes, et les statistiques en termes de nombre de fois où ces groupes ont été observés.

Enfin, un troisième fichier contient des indicateurs tels que le temps de convergence (pour certaines simulations de test où l'on présente de nombreuses fois un même motif afin d'observer si la représentation se stabilise), et la cohérence et la discrimination, qui sont deux métriques ad hoc que nous proposons. La cohérence représente le fait qu'un même motif en entrée est classé dans un même groupe. La discrimination représente le fait qu'un même groupe ne contient qu'un seul motif. Nous ne prétendons aucunement que ces deux indicateurs devraient toujours être vérifiés, ce n'est probablement le cas que dans une minorité de simulations, mais cela nous a servi, là encore, à mieux comprendre les résultats de nos simulations dans certains cas de test.

Armés de ces différentes sorties, nous avons dans un premier temps débogué notre implémentation de façon à obtenir un résultat visuellement cohérent dans notre interface graphique, puis nous avons essayé de valider le fonctionnement de notre implémentation de HTM-CLA dans quelques cas de tests. Ces tests ont pris la forme de tests de partitionnement, puisque ce test nous semble être extrêmement révélateur de la représentation construite par la hiérarchie. Malheureusement, nous n'avons pas réussi à obtenir un résultat satisfaisant. Par exemple, nous constatons toujours des oscillations de représentation même lors d'une longue séquence d'entrée où le même motif est toujours présenté. Un exemple de telle simulation est présenté dans l'annexe E.

Nous nous sommes alors heurtés à la vraie difficulté d'implémentation des algorithmes de type réseau de neurone : il est difficile de comprendre le lien entre le fonctionnement d'une unité et le fonctionnement global, qui relève plutôt d'un phénomène d'émergence. Ainsi, nous n'avons pas pu identifier de façon sûre la cause de ces instabilités de représentation que nous observions. Il est possible qu'elles viennent tout simplement d'un mauvais réglage des différents paramètres que nous avons détaillés plus haut. Toutefois, nos tentatives de les faire varier manuellement n'ont pas encore permis de trouver un jeu de paramètres aboutissant à une représentation stabilisée.

Malheureusement, cet échec nous a conduit à arrêter là nos tests, puisque sans implémentation fonctionnelle du moteur parvenant à une représentation stable, il était illusoire de tester nos différents algorithmes et d'espérer obtenir de bons résultats, a fortiori sur des jeux de données réelles intéressantes mais plus bruitées.

Toutefois, bon nombre de nos propositions sont en fait complètement indépendantes de l'implémentation choisie. On peut donc imaginer les adapter à un autre implémentation fonctionnelle de HTM-CLA afin de pouvoir tester leur pertinence. Notamment, Numenta a rendu publique une version de son algorithme assez récemment, alors que nous rédigeons déjà ce mémoire. Cette perspective sera détaillée en partie 6.3.3.

4.3.5 Proposition d'utilisation de HTM-CLA pour la génération de comportement

Nous pensons que l'automatisation d'une partie des tâches des tuteurs et responsables de formation est très importante pour les aider à effectuer leur travail de suivi de façon plus efficace. Une telle automatisation est même cruciale pour les cours à très grande échelle, où le taux d'encadrement ne peut jamais être adéquat. Dans le chapitre 5, nous parlerons de l'automatisation des tâches d'import de données et du rôle joué par notre application, GIGA, en leur permettant de trouver plus aisément les données nécessaires à leur prise de décision. Toutefois, nous pensons également à un autre aspect que nous souhaitons aborder plus en détail ici.

Nous pensons en effet pouvoir également automatiser une partie du retour de conseils envers les étudiants dans certaines situations pré-définies. Cette automatisation pourrait aussi prendre la forme d'un tuteur virtuel intelligent évoluant et agissant sur le LMS en interaction directe avec les élèves et leurs formateurs. Ce système pourrait guider les élèves en temps réel sur les plate-formes de cours tout en faisant remonter des informations en temps réel aux tuteurs si elles sont jugées importantes.

Son domaine d'action pourrait être, par exemple, de suggérer à un élève d'aller lire la leçon associée à un quiz auquel ils viennent d'échouer, ou à l'inverse de lui recommander un quiz s'il vient de finir de lire la leçon associée, ou de le complimenter et lui suggérer d'aller voir la leçon suivante s'il a bien réussi suffisamment d'activités. Il pourrait également envoyer un e-mail à un élève qui s'est peu connecté pour lui suggérer de retourner sur la plate-forme, et en prévenir le tuteur ; ou prévenir les élèves de l'ouverture d'un nouveau cours ou d'une nouvelle activité, ou de l'approche d'une deadline.

4.3.5.1 Tuteur virtuel intelligent

Un tuteur virtuel intelligent (en anglais *intelligent tutoring system*, ou ITS) [Paviotti et al., 2012] cherche à fournir automatiquement à l'élève des instructions et des retours personnalisés et immédiats. Pour atteindre leur but, ils sont souvent conçus selon des préceptes issus des théories de la psychologie cognitive de l'apprentissage pour mieux modéliser le raisonnement de l'élève, et peuvent utiliser des méthodes d'intelligence artificielle. Il s'agit donc d'un système évoluant dans un environnement virtuel et interagissant avec les élèves.

Par exemple, une équipe de recherche de l'université américaine Carnegie Mellon a proposé un programme nommé Cognitive Tutors [Koeclinger et al., 2000] et a montré que les élèves qui l'utilisaient avaient un score amélioré de 15% à 85% suivant les types de problème.

Un tuteur intelligent peut être créé par une simple liste de règles prédéfinies. Toutefois, ses interventions sont alors limitées aux cas de figure pensés par ses concepteurs, et dépendent généralement de seuils plutôt que d'avoir un fonctionnement un peu continu. C'est pourquoi nous jugeons plus intéressant de lui faire prendre la

forme d'un moteur de génération de comportement intelligent.

4.3.5.2 Génération de comportement

La discipline de la génération de comportement, aussi appelée de la sélection d'action, cherche à animer des créatures autonomes fonctionnant dans un environnement virtuel et bien adaptées à cet environnement. Ces créatures peuvent également être appelées des agents intelligents.

Si l'environnement et les objectifs du projet s'y prêtent, un des buts de cette discipline peut être le réalisme, comme par exemple, si l'on cherche à créer un humain virtuel capable d'interaction avec un humain réel, ou une créature capable de reproduire les comportements d'une créature réelle. Parmi les domaines utilisant des simulations de comportement réalistes, on peut lister l'industrie du film (pour la simulation de foule ou de créatures imaginaires) et des jeux vidéo, mais également l'industrie du tourisme, qui cherche à faire des guides virtuels, et l'éducation, plutôt dans le domaine des jeux sérieux et des simulations.

On peut utiliser deux sortes d'approches pour la génération de comportement. L'approche computationnelle cherche à modéliser les connaissances de l'agent par une représentation symbolique et à le faire agir en conséquence. Elle utilise souvent des machines à état, des systèmes experts, et des algorithmes de planification. La deuxième approche est celle de la vie artificielle, qui cherche à ressembler aux créatures vivantes réelles, parfois par des algorithmes d'apprentissage, comme les systèmes de classeurs, et d'évolution, leur permettant souvent une plus grande adaptabilité. C'est notamment le cas des créatures de Karl Sims [Sims, 1994] ou des boïds de Craig Reynolds [Reynolds, 1987].

Il existe trois grandes familles d'agents autonomes. Les agents délibératifs sont caractérisés par une grande intentionnalité de leurs actions ; leurs perceptions sont analysées par un modèle cognitif de leur environnement, puis un module de planification calcule à partir de ce modèle une suite d'actions leur permettant d'atteindre leurs buts. En contrepartie, ils nécessitent beaucoup de ressources de calcul et peuvent être lents à réagir.

Les agents réactifs sont généralement constitués de différents modules autonomes agissant en parallèle, arbitrés par un système qui permet de déterminer laquelle des actions proposées est la plus importante à un moment donné, ce qui leur permet par exemple de réagir instantanément à un danger. On retrouve aussi dans cette famille les agents issus du domaine de la vie artificielle et notamment ceux qui utilisent des mécanismes évolutionnaires pour atteindre un comportement optimal. Ces mécanismes sont généralement appliqués sur des réseaux de neurones ou des systèmes de classeurs. Cette approche a d'abord été proposée par Gibson [Gibson, 1979a].

Finalement, les agents hybrides combinent une couche réactive et une couche délibérative par le biais d'un séquenceur. Cela leur permet de combiner des capacités de planification sur le long terme avec des capacités de réaction extrêmement rapides. Ils sont donc extrêmement prometteurs. Ils s'inspirent de la théorie des frames de Marvin Minsky [Minsky, 1988].

Les systèmes de classeurs [Holland and Reitman, 1978] sont dédiés à l'apprentissage de règles de type condition \rightarrow action. Cet apprentissage se fait par algorithme génétique. La base de règle ainsi constituée forme le générateur de comportement. Dans l'équipe VORTEX, de nombreux travaux ont été basés sur ces systèmes. Cédric Sanza [Sanza, 2001] puis Olivier Héguay [Héguay, 2003] ont utilisé des classeurs pour produire des agents capables d'élaborer des stratégies dans le cadre de jeux de basket ou de football virtuels.

Stéphane Sanchez [Sanchez, 2004] a proposé une architecture hybride intitulée ViBes (pour *Virtual Behaviours*), qu'il a utilisée pour obtenir des comportements très réalistes d'agents humanoïdes dans un environnement virtuel. Cette architecture comporte notamment un module de connaissances permettant à l'agent de manipuler les objets qui l'entourent et de se souvenir où il les a déjà vus lors de son exploration de l'environnement. Ensuite, différents modules comportementaux, réactifs et cognitifs, entrent en compétition pour l'accomplissement des buts de l'agent.

4.3.5.3 Génération de comportement avec HTM-CLA

Nous proposons d'utiliser l'algorithme HTM-CLA comme moteur de base d'un système de génération de comportement, qui lui-même servirait de base à la création d'un tuteur virtuel intelligent adapté au suivi des élèves.

Dans la partie 4.2.4, nous avons parlé du point de vue de l'intelligence artificielle écologique, qui soutient que l'intelligence et la complexité du comportement ne peuvent venir que d'un environnement également complexe. Ce point de vue est bien résumé par la citation suivante :

« Une fourmi, considérée comme un système comportemental, est très simple. La complexité apparente de son comportement au fil du temps est essentiellement un reflet de la complexité de l'environnement dans lequel elle se trouve. »

— Herbert Simon, *The Sciences of the Artificial* (traduction personnelle)

Nous avons également expliqué en quoi HTM-CLA, basé sur le paradigme mémoire-prédiction, est conforme à ce paradigme, puisqu'il se base sur les régularités spatiales et temporelles de l'environnement complexe et permet donc de le modéliser plus efficacement. Il est compréhensible qu'un comportement basé sur un modèle plus réaliste de l'environnement soit également mieux adapté. C'est ce qui justifie notre intérêt pour cette idée.

4.3.5.4 Adaptive Memory-Prediction Framework

Une autre équipe a tenté de réaliser ce genre de travaux : Rawlinson et Kowadlo ont proposé Adaptive Memory-Prediction Framework (AMPF) [Rawlinson and Kowadlo, 2012], qui ajoute une couche comportementale à une implémentation du paradigme mémoire-prédiction basée sur l'utilisation d'une carte auto-organisatrice pour représenter chaque région.

Selon eux, les actions sont présentes dans toute la hiérarchie, avec aux niveaux inférieurs les actions basiques, et aux niveaux supérieurs des macro-actions abstraites qui durent plusieurs pas de temps. Ils proposent de présenter l'action effectuée au pas de temps précédent comme une entrée au bas de la hiérarchie, dans une région ad hoc qui fournira alors comme sortie prévue l'action suivante à effectuer, après la passe descendante de l'algorithme.

Ils choisissent d'utiliser une méthode d'apprentissage par renforcement, qui leur paraît plus adaptée à l'évaluation de situations complexes. Pour ce faire, ils ajoutent entre chaque niveau hiérarchique un nouveau composant qu'ils appellent le corrélateur de récompense, et qui motive l'agent à changer de décision si la précédente était mauvaise, en calculant la dérivée de la récompense. Pour cela, il modifie la prédiction qui va être réalisée. En effet, le mécanisme de prédiction est le même que le mécanisme de sélection d'action. Cet élément ne peut influencer que les éléments qui étaient actifs au pas de temps précédent, qui sont censés avoir influencé le choix de l'action.

Ils ont testé les résultats de leur modèle sur le problème pierre-feuille-ciseaux, où l'adversaire de leur agent était un modèle extrêmement simple qui alternait entre les trois actions à tour de rôle, étant ainsi très prévisible. Sans le corrélateur de récompense, leur modèle parvenait à prédire les actions de l'adversaire, mais ne cherchait pas à les contrer. En activant ce composant, ils sont parvenus à un taux de réussite de 93%.

4.3.5.5 Le modèle comportemental Cortexionist

Nous allons d'abord faire un petit détour par un modèle assez proche de ce que nous voulons faire et issu des travaux d'un chercheur de notre équipe. Dans sa thèse [Panzoli, 2008], David Panzoli a développé l'architecture Cortexionist pour donner à des créatures artificielles une intelligence comportementale.

Cortexionist représente une architecture hybride. Toutefois, contrairement au choix de la plupart des systèmes hybrides, Panzoli a choisi une approche entièrement connexionniste, plus cohérente avec le domaine de la vie artificielle. Dans son cas, les deux parties, cognitive et réactive, sont créées par un réseau de neurones ad hoc.

Le contrôleur utilisé par un agent utilisant l'architecture Cortexionist est un réseau de neurones comprenant deux couches d'architecture très différente, ce qui fait qu'il peut être vu sous deux angles. Verticalement, il s'agit d'un perceptron, qui relie une couche d'entrée à une couche de sortie. Mais cette couche d'entrée, vue horizontalement, est connectée comme un réseau associatif, rassemblant les perceptions issues des différents capteurs de l'agent. Panzoli fait l'analogie de ces deux vues avec une mémoire procédurale et un comportement réactif issu du cerveau reptilien (vue verticale) et une mémoire sémantique et un comportement plus réfléchi issu du néocortex (vue horizontale). Cette architecture est présentée dans la figure 4.20.

Au sein de la couche associative, une loi d'apprentissage de type hebbien assure l'apprentissage et la création de connaissances. Une relation d'inhibition y est rajoutée pour les neurones d'un même voisinage sensoriel qui ne s'activent pas en même temps.

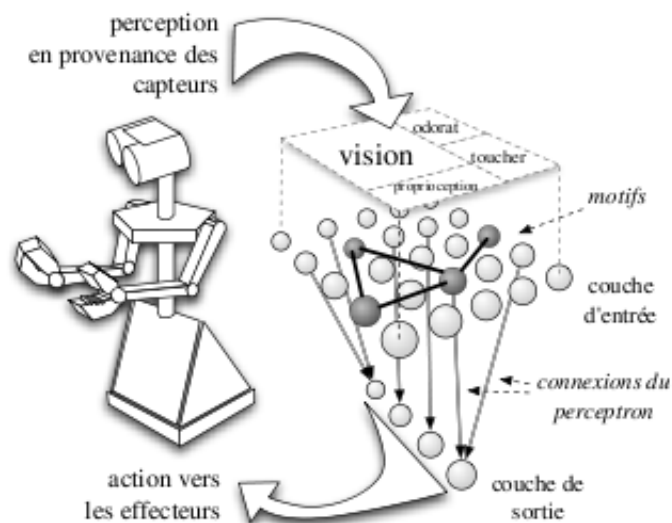


FIGURE 4.20 – L'architecture Cortexionist

Le système de calcul des poids de la connexion assure la complétion des motifs incomplets au-delà d'un certain seuil de neurones activés.

Afin d'apprendre le choix d'une action, le modèle doit faire avec deux mécanismes d'apprentissage très différents. Celui qui est le mieux adapté au réseau associatif est un apprentissage non supervisé, non incrémental et en ligne ; au contraire, pour le perceptron, un apprentissage supervisé, incrémental et hors ligne semble plus approprié.

Le fonctionnement de ce réseau est asynchrone. Le modèle de neurone choisi est un modèle relativement proche d'un neurone biologique, avec notamment un potentiel d'action passant par des phases de dépolarisation, repolarisation et hyperpolarisation.

Tests en environnement virtuel Les tests de l'architecture Cortexionist ont été menés dans un environnement de simulation virtuel. L'agent est une créature qui possède une mère et des congénères, mais aussi un prédateur. Toutes les autres créatures que l'agent ont un comportement pré-programmé. Une interface graphique créée pour en visualiser les résultats est présentée dans la figure 4.21.

L'agent a deux sens, la vue et l'odorat, et peut se déplacer dans son environnement avec deux comportements pré-programmés, le rapprochement et la fuite, ce dernier étant déclenché par la trop grande proximité du prédateur. Il reconnaît également dès le début de la simulation l'odeur de sa mère. Il doit aussi gérer son niveau d'énergie, qui baisse en proportion de la vitesse à laquelle il se déplace, elle-même en relation directe avec la distance à laquelle le prédateur est détecté. Le contrôleur de l'agent contient donc au final les neurones suivants : dans la couche associative, les perceptions couleur verte, couleur pourpre, tentacule, bras, odeur de la mère et proximité ; dans les actions possibles, se rapprocher ou s'enfuir.

Dans les tests, le comportement de l'agent est comparé avec celui d'un agent



FIGURE 4.21 – L'environnement de test pour Cortexionist

contrôlé par un simple perceptron, sans réseau associatif dans la couche d'entrée.

Dans chacune des expériences (avec ou sans la mère et le congénères), le comportement de l'agent contrôlé par l'architecture Cortexionist est meilleur, puisqu'il conduit l'agent à mieux gérer son taux d'énergie et à mieux savoir reconnaître et fuir le prédateur tout en cherchant protection auprès de sa mère et ses congénères. Ainsi, l'agent Cortexionist évite d'être tué par le prédateur.

Discussion L'originalité de ces travaux se situe dans le choix de l'absence de modélisation manuelle de comportements intelligents variés, qui résulte en une approche dynamique et adaptable. Le parti pris de l'auteur était de prouver que l'intelligence d'un agent peut être issue de la façon dont il mémorise des connaissances à partir de l'exploration de son environnement, et que la complexité du comportement vient de la complexité de l'environnement.

On peut reprocher à cette architecture de n'avoir été testée que sur un domaine de très petite taille (six neurones dans la couche d'entrée et deux dans la couche de sortie). Malgré cela, les résultats obtenus dans ces expériences sont très encourageants. Le seul problème qui a été remarqué lors de ces expériences relève du compromis bien connu à faire entre stabilité des connaissances et plasticité du réseau.

De plus, lesdits neurones ne correspondaient pas à des concepts ou actions élémentaires, mais bien de plus haut niveau, comme par exemple "s'enfuir". Néanmoins, on pourrait, pour augmenter le réalisme du modèle, utiliser des niveaux de connaissances hiérarchiques ; les perceptions à l'entrée d'une couche seraient donc les connaissances en sortie de la suivante. A priori, les concepts de haut niveau créés seraient similaires à ceux choisis dans l'expérience.

L'auteur propose, pour améliorer son modèle, le rajout de diverses couches associatives permettant un comportement anticipé grâce à des neurones dans les couches inférieures ayant une plus longue durée d'activation ainsi que des connexions plus longues et faisant donc intervenir un retard lors de la transmission de l'activation, permettant ainsi à des neurones activés à des intervalles plus grands de devenir reliés dans les couches inférieures. Ce mécanisme est assez proche du mécanisme proposé par Hawkins passant par le thalamus pour introduire un retard, même si le retard ne sert pas dans les HTM à étendre la prédiction, mais plutôt à repérer les changements

d'état.

4.3.5.6 Un modèle comportemental basé sur HTM-CLA

Nous proposons de nous inspirer des travaux de Panzoli pour proposer un modèle comportemental basé sur HTM-CLA.

Notre proposition Nous avons présenté deux modèles permettant de générer un comportement intelligent à partir d'un modèle de représentation de connaissances inspiré par le cerveau humain. Ces deux modèles prouvent la faisabilité d'un mécanisme de génération de comportement adaptatif et capable d'anticiper l'avenir, basé sur HTM.

Dans AMPF, ce modèle est inspiré de HTM, que nous pensons être le meilleur choix de par ses capacités de prédiction et de mémorisation de séquences temporelles, qui lui permettent de mieux comprendre son environnement et de faire preuve d'un comportement plus complexe, anticipant mieux l'avenir. Toutefois, le mécanisme du corrélateur de récompense permettant un apprentissage par renforcement ne convient pas aux applications que nous avons en tête. En effet, nous ne savons pas comment évaluer une action de retour sur les élèves. L'idée de Cortexionist d'étendre un comportement instinctif ou réactif de façon intelligente nous paraît bien plus adaptée, puisqu'elle permet une sorte d'apprentissage par l'exemple.

Il paraît important à Rawlinson et Kowadlo de ne pas dupliquer les connaissances en dehors du système, et c'est pourquoi ils choisissent un système de génération de comportement intégré à la hiérarchie. De même, le contrôleur de Panzoli est une partie du même réseau de neurones qui stocke les connaissances. C'est aussi le choix que nous faisons, avec même encore davantage d'intégration que Rawlinson et Kowadlo, puisque nous ne rajoutons pas de composant à la hiérarchie, et nous respectons les niveaux pré-définis. En outre, cette intégration peut avoir à terme un autre avantage : celle d'en faire un seul et unique système avec le HTM qu'on souhaite intégrer à GIGA pour les applications plus classiques, permettant ainsi une réutilisation encore plus importante. Par exemple, le même HTM pourrait lever une anomalie en examinant les logs et décider d'envoyer un e-mail à un tuteur pour l'en prévenir.

Nous projetons donc de concevoir et créer un système capable d'utiliser les connaissances stockées dans une hiérarchie de type HTM-CLA sous forme de motifs spatiaux assemblés en séquences temporelles pour étendre des comportements simples et pré-définis. Nous pensons que Rawlinson et Kowadlo ont raison en faisant ré-entrer les actions en bas de la hiérarchie et souhaitons reprendre cette idée.

Nous pensons réutiliser le domaine de test de Rawlinson et Kowadlo (pierre-feuille-ciseaux) dans un premier temps pour comparer nos résultats aux leurs. Il s'agit d'un problème où l'apprentissage par exemple peut fonctionner, en donnant comme exemple au système les mouvements gagnants (pierre si l'adversaire joue ciseaux, ciseaux s'il joue papier et papier s'il joue pierre). Nous espérons obtenir un taux de succès comparable au leur sur leur adversaire basique, mais également

introduire des adversaires avec une stratégie moins prévisible, et maintenir face à eux un taux de succès supérieur à celui obtenu en jouant au hasard (33%).

Nous avons également choisi un autre domaine de test, plus utilisé dans le domaine de la génération de comportement, notamment dans la communauté des systèmes de classeurs, mais restant assez simple, à savoir Maze et Woods. Le but de ces problèmes est d'atteindre la nourriture dans un environnement présentant des obstacles. Quelques exemples de problèmes relevant de ces domaines sont présentés dans la figure 4.22. Il s'agirait donc ici de faire apprendre une carte donnée au système, pour qu'il la parcoure ensuite de façon optimale, en se dirigeant vers la nourriture par le plus court chemin possible. Par rapport au jeu pierre-feuille-ciseaux, la différence est l'ajout d'une dimension spatiale et pas seulement temporelle au problème. Il s'agit aussi d'un problème où l'apprentissage par exemple peut fonctionner, en donnant au système l'exemple de se diriger vers la nourriture quand il est juste à côté, et de contourner un obstacle.

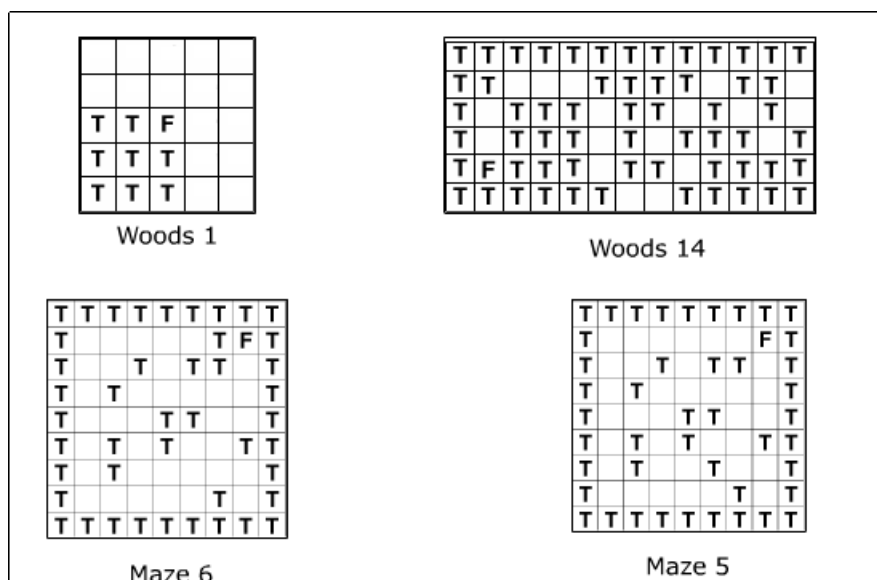


FIGURE 4.22 – Quelques exemples des problèmes Maze et Woods. T représente un arbre et F la nourriture (des mots anglais *tree* et *food*).

Si l'on suppose l'existence d'une version de HTM capable de comportement, on peut s'attendre à voir l'apparition de comportement abstrait dans les niveaux supérieurs de la hiérarchie, avec une abstraction à la fois spatiale et temporelle : ainsi, les comportements élémentaires fournis comme exemples et présents tout en bas de la hiérarchie « me déplacer d'une case vers la nourriture si elle est à côté de moi » et « éviter les obstacles » et « continuer mon déplacement si je n'ai pas atteint la nourriture » peuvent donner le comportement abstrait présent plus haut « me déplacer sur plusieurs cases en évitant les obstacles jusqu'à atteindre la nourriture ». Il s'agit donc d'un système de contrôle hiérarchique.

Toutefois, comme l'avaient déjà noté Rawlinson et Kowadlo, toute forme d'ap-

prentissage de type pré-câblage, comme celle que nous proposons, pose un problème dans HTM, qui est celui de savoir situer les connaissances dans la hiérarchie et même au sein des régions. Selon eux :

« Avant l'apprentissage, il n'est pas possible de prédire quels concepts objectivement utiles existeront, ni à quel endroit on pourra les trouver dans la hiérarchie. [...] Bien qu'il soit nécessaire que les concepts possédant les abstractions et les invariances désirées existent quelque part dans la hiérarchie, il n'est pas désirable de devoir définir où avant l'apprentissage. ».

Panzoli n'avait pas ce problème puisque le contrôleur de Cortexionist était un réseau de neurones ad hoc où chaque neurone avait dès sa création une signification. Ainsi, pour nous, la difficulté sera plus grande que pour Panzoli, puisque nous ne pourrions montrer facilement que des exemples de comportement avec très peu d'abstraction, à bas niveau. Ces exemples n'auront pas nécessairement une association directe avec le comportement souhaité : on ne peut que montrer l'exemple de quitter une case sans percuter un obstacle, mais pas celui de quitter une case en allant vers la direction générale de la nourriture. Nous serons donc très intéressés de découvrir si le comportement observé est celui d'une prise de chemin optimale vers la nourriture, ou si les exemples ne sont pas suffisants pour cela. Si ce n'est pas le cas, nous pourrions être amenés à créer une forme d'apprentissage hybride, utilisant exemples et renforcement.

Pour notre application de tuteur virtuel intelligent, on pourrait fournir des exemples de suivi appliqué à un élève donné dans une situation donnée, et voir si notre système arrive à l'étendre aux autres élèves, ainsi qu'aux autres gammes de notes et d'activité.

Pour résoudre le problème de trouver les connaissances, nous proposons un apprentissage en deux temps, analogue à celui proposé par Panzoli, mais inversé par rapport à lui, puisque nous n'avons pas l'avantage de son pré-câblage.

La première phase de l'apprentissage se fait sans comportement pré-câblé et vise uniquement à construire la représentation du reste de l'environnement. Pour cette raison, on désactive l'apprentissage de la région qui prend les actions comme entrées (et qui en fournit les sorties). On laisse l'apprentissage classique de HTM-CLA se faire, de façon non supervisée, tout en explorant la région en proposant des actions faisant changer l'environnement, jusqu'à obtenir une convergence de la représentation. Ces actions peuvent être choisies aléatoirement ou pour présenter une suite d'actions optimale (par exemple, pour Maze, il suffit de connaître la carte et la situation initiale pour proposer un chemin optimal jusqu'à la nourriture). Nous testerons ces deux méthodes pour comparer leur efficacité.

La deuxième phase de l'apprentissage consiste à rajouter des exemples de comportement. Ces exemples prennent la forme d'un couple (situation, action). Pour les faire apprendre, nous réactivons l'apprentissage de la région qui prend les actions comme entrée. Nous activons l'aspect situation du couple dans le reste de la hiérarchie, puis au pas de temps suivant, nous faisons entrer l'action choisie dans cette région. Ainsi,

c'est cette action que le mécanisme de prédiction apprendra à prédire pour le pas de temps suivant dans cette situation. Or, comme le mécanisme de sélection d'action est justement le mécanisme de prédiction, c'est cette action qui sera choisie dans cette situation. Là encore, on répète l'apprentissage jusqu'à la stabilisation de la représentation dans cette région.

Un des inconvénients de notre proposition est qu'elle contredit les théories modernes sur l'importance de contrôler son action pour comprendre le changement de perception qu'elle induit. Toutefois, nous ne voyons actuellement pas de possibilités de se conformer à cette théorie étant donné la limitation sur le pré-câblage que nous venons d'expliquer.

Si toutefois cette approche s'avérait un échec, à cause des écueils possibles que nous avons mentionnés, nous avons également une autre proposition qui pourrait permettre de pré-câbler des comportements abstraits. Il s'agira de repérer les motifs abstraits présents dans la hiérarchie par la même méthode que nous comptons utiliser pour proposer des sorties de type inférence ou explication à HTM-CLA, puis d'utiliser ces motifs pour proposer des exemples de comportement abstrait. Nous ne savons pas encore si les actions associées à ces exemples pourront être des actions abstraites situées haut dans la hiérarchie, ou si les motifs de comportement abstrait ne conviendront pas à nos besoins, auquel cas il nous faudra associer des actions élémentaires à des situations abstraites.

Nous avons d'ailleurs, dans un premier temps, proposé un premier modèle de génération de comportement basé sur des méthodes assez similaires aux algorithmes que nous avons retenus pour le partitionnement et la classification, avec un simple dictionnaire qui associait les motifs activés dans la plus haute région avec une action. Ce système était néanmoins extrêmement basique, puisqu'il n'offrait guère de possibilité de création de comportement abstrait. Nous avons présenté cette proposition lors des journées de l'AFRV 2010 [Bovo et al., 2010].

Implémentation Nous présentons en annexe B des détails d'implémentation de nos propositions. Nous proposons à la figure B.5 un modèle qui, comme souhaité, intègre les actions élémentaires en bas de la hiérarchie. Nous présentons également une première démonstration de l'instanciation sous forme de classes pour nos deux domaines de test, Maze/Woods et pierre-feuille-ciseaux, dans la figure B.6.

Nous avons également proposé une interface de visualisation adaptée aux problèmes de type Maze/Woods, qui étend l'interface proposée précédemment. Elle est visible à la figure 4.23. Nous y avons rajouté l'affichage du problème Maze/Woods ainsi que de la région d'entrée associée. Nous avons lancé plusieurs simulations sur un même problème grâce à l'interface de contrôle. Le résultat de ces simulations est présenté dans un graphe qui affiche le nombre de coups mis avant de trouver la sortie.

Nous n'avons pas réellement cherché à dépasser le stade de l'implémentation pour aller à celui des tests, dans la mesure où nous savions que notre moteur HTM comportait encore des problèmes. Toutefois, là encore, nos propositions sont

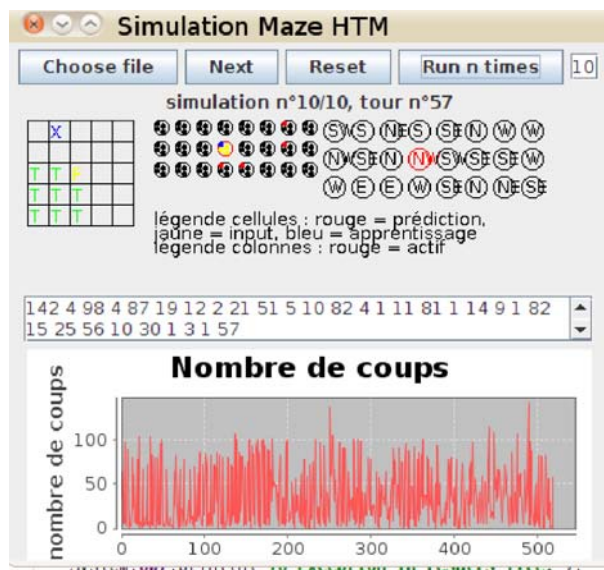


FIGURE 4.23 – Notre interface de visualisation des problèmes de type Maze/Woods

génériques et capables de s'adapter à n'importe quelle implémentation de HTM-CLA, et pourraient donc être testées et validées sur une autre implémentation.

Nous avons maintenant terminé l'inventaire de nos propositions. Séparées sur les trois chapitres précédents, elles regroupent donc des indicateurs adaptés au suivi des élèves, des méthodes de suivi plus poussées par apprentissage automatique, avec des méthodes classiques mais également l'idée d'utiliser HTM, un algorithme innovant et bien adapté, et l'idée d'utiliser HTM comme base de génération de comportement pour la création d'un tuteur virtuel intelligent. Nous allons maintenant montrer comment nous avons mis en pratique ces propositions avec la création d'une application web destinée au suivi des élèves.

Chapitre 5

Proposition d'application dédiée au suivi des étudiants

GIGA est une application Web de suivi d'élèves en formation e-learning, que nous avons collaboré à concevoir et à réaliser en partenariat avec l'entreprise Andil. Le sigle GIGA signifie « Gestionnaire d'Indice Général d'Apprentissage ». En effet, cette application est entre autres basée sur notre proposition d'indicateurs pertinents pour le suivi d'étudiants. En regroupant les propositions des chapitres précédents, nous obtenons une proposition d'application complète dédiée au suivi des étudiants. Cette application utilise les données et indicateurs définis dans le chapitre 2, et les méthodes d'apprentissage automatique proposées dans le chapitre 3. Notre proposition d'utilisation de HTM-CLA pour l'apprentissage automatique, présentée dans le chapitre 4 est également incluse dans cette application.

5.1 Architecture

Nous allons voir ici comment l'architecture de cette application a été conçue.

Il existe différents points de vue possibles permettant de décrire l'architecture d'une application. On peut notamment distinguer :

- l'architecture métier, qui permet de mieux comprendre comment les différents métiers impliqués interagissent entre eux. On cherche à comprendre le projet dans le cadre de l'activité et des processus de l'entreprise.
- l'architecture fonctionnelle, qui raisonne en terme des fonctionnalités de l'application et des liens entre ces fonctionnalités.
- l'architecture applicative, qui fait apparaître des composants techniques et des fonctionnalités, ainsi que des regroupements à l'échelle applicative.
- l'architecture technique, qui fait les choix techniques d'outils et d'implémentation, et s'intéresse à la problématique du déploiement.

Nous avons choisi de ne pas présenter ici d'architecture métier, car ce point de vue est plus adapté aux gros projets devant s'intégrer dans une grande entreprise dotée d'un existant fort.

5.1.1 Cahier des charges

Le cahier des charges de cette application a été partiellement décrit dans notre introduction et des détails y ont été ajoutés dans les parties suivantes, notamment avec notre liste de métriques permettant le suivi des élèves et notre souhait émis d'utiliser les possibilités de l'exploration de données et de l'apprentissage automatique.

Nous souhaitons toutefois y rajouter des desiderata de l'ordre de l'utilisabilité de cette application, qui n'est pas destinée à un public académique mais bien à des tuteurs et responsables de formation. Les personnes que nous avons contactées devaient jusqu'à présent recueillir et traiter les données manuellement : recopier dans un tableur les données fournies par l'outil de statistiques de Moodle pour avoir une vue d'ensemble, repérer les élèves en difficulté, interroger les enseignants pour avoir leur avis sur les élèves, envoyer des mails et téléphoner aux élèves et aux enseignants pour leur fournir des retours... Nous souhaitons leur faciliter ce processus.

Nous nous sommes fixé comme but de centraliser les données fournies par une formation en e-learning. Actuellement, beaucoup de données sont dispersées en deux principaux endroits : les traces d'activités sont présentes sur le LMS (Moodle ici) et d'autres données (administratives, d'activité en présentiel, d'historique de contact et de communication, de notes aux examens finaux) sont conservées par les responsables de formation et leur équipe administrative ainsi que par les divers intervenants et enseignants, parfois dans des solutions peu adaptées comme des tableurs Excel. Nous voulons donc rassembler, autant que possible, les données pédagogiques pertinentes en un seul endroit de stockage. Cette centralisation nous permettra également de pouvoir comparer plusieurs types de LMS, en s'affranchissant de leur format spécifique de stockage.

Notre deuxième but vise à automatiser le plus de tâches possibles parmi celles qui sont actuellement faites par des humains comme les responsables de formation. L'import des données mentionné plus haut doit également se faire de façon entièrement automatique.

Nous n'avons pour l'instant pas prévu d'approche réflexive visant à donner directement à l'étudiant accès à son positionnement pour qu'il choisisse lui-même comment réagir, mais cela pourra être étudié si jugé pertinent par les responsables de formation. Cela peut poser problème dans le sens où, comme beaucoup de nos données sont relatives à un positionnement dans le groupe, l'étudiant aurait alors nécessairement accès à beaucoup de données concernant ses camarades, auxquelles ces derniers n'auraient pas forcément envie de lui donner accès. Notre application est donc pour l'instant uniquement à destination des tuteurs et responsables de formation.

Enfin, ces données doivent être extrêmement faciles à lire et à comprendre, avec différents niveaux de granularité, des vue individuelles et globales et un caractère très visuel.

On peut résumer ainsi notre cahier des charges :

- centraliser toutes les données associées à une formation en ligne
- réaliser un suivi des élèves à l'aide des métriques proposées et des méthodes

- d'exploration de données et d'apprentissage automatique
- présenter ces données aux responsables de formation de façon aisément compréhensible
- être le plus à jour possible (minimum une mise à jour quotidienne) avec un import de données automatique

5.1.2 Architecture fonctionnelle

Il s'agit à cette étape de prendre en compte le cahier des charges de notre application, d'en tirer un découpage de fonctionnalités et de formaliser le lien entre elles. La description de l'architecture peut déjà effectuer un certain nombre de choix pré-conception s'ils sont nécessaires à la réalisation de cette analyse.

Les fonctionnalités que nous avons trouvées sont représentées dans le graphe 5.1. On y retrouve des fonctionnalités liées aux trois aspects habituellement trouvés dans beaucoup d'applications, à savoir le stockage de données, le cœur de métier et l'interface utilisateur.

Par exemple, pour notre projet, les données que nous avons présentées au chapitre 2 doivent être regroupées et stockées, et on doit pouvoir y accéder, ce qui est assez classique. S'y ajoute une fonctionnalité d'import régulier et automatique des données venant du LMS, et d'import manuel des données venant d'autres sources. Notre cœur de métier est représenté par le calcul des indicateurs et des différentes sorties des méthodes d'apprentissage automatique que nous avons définies dans le chapitre 3. Les fonctionnalités liées à l'interface ont été présentées dans le cahier des charges que nous venons d'établir.

Quelques fonctionnalités supplémentaires sont un peu transversales. Ainsi, le filtrage des données est sollicité par l'utilisateur, mais va affecter les calculs réalisés. La personnalisation de l'interface, par la définition des seuils déterminant le code couleur, est demandée par l'utilisateur, mais va ensuite être mémorisée dans notre base de données. De même, la personnalisation de la pondération des différents indicateurs pour le calcul de ceux de niveau hiérarchique supérieur va affecter tous les aspects de notre application.

5.1.3 Architecture applicative

Dans une architecture applicative, on ne retrouve pas de composant fonctionnel métier, mais bien des composants techniques qui supportent les composants fonctionnels métier. Ils sont regroupés au sein d'applications. Des services applicatifs transversaux viennent compléter le schéma.

On y retrouve généralement, en plus des fonctionnalités, une liste du matériel utilisé, du réseau reliant ces matériels, des méthodes de transfert et de gestion d'information, et des services transverses.

Ce découpage aide à passer des couches fonctionnelles aux niveaux de distribution de l'application.

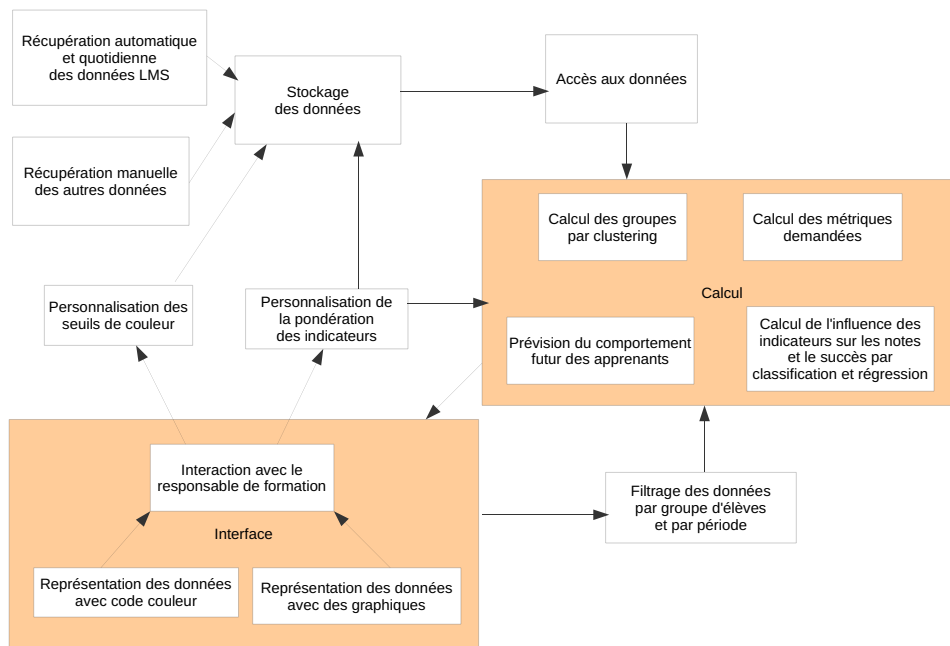


FIGURE 5.1 – Notre proposition d'architecture fonctionnelle pour GIGA

Architecture à trois couches L'architecture à trois couches est l'application du modèle plus général qu'est l'architecture multi-couches.

Un composant logiciel est une partie plus ou moins fonctionnellement indépendante d'un logiciel. Ainsi, un moteur de calcul, une base de données, une bibliothèque ou un framework sont des composants logiciels.

La division en couches consiste à regrouper les composants possédant une grande cohésion de manière à créer un empilement de paquetages de composants ; tous les composants des couches supérieures dépendent fonctionnellement des composants des couches inférieures.

Dans ce modèle, l'architecture logique du système est divisée en trois couches :

- la couche de présentation des données (affichage, IHM, dialogue avec l'utilisateur) ;
- la couche métier (traitement des données, logique applicative) ;
- la couche d'accès aux données persistantes, souvent via une base de données.

Chaque couche propose un ensemble de services rendus, qui sont mis à disposition de la couche supérieure, et ne communique qu'avec ses voisins immédiats.

Nous avons retenu ce modèle d'architecture qui correspond à nos besoins. En effet, nous avons besoin de chacune de ces fonctionnalités, et les découpler permet de faire évoluer chacune plus facilement. La proposition d'architecture applicative correspondante est résumée dans le schéma de la figure 5.2. Nous y retrouvons la

plupart de nos fonctionnalités réparties entre ces trois couches.

Dans ce schéma, nous avons également représenté, en plus de l'application, qui est distribuée sur un serveur informatique et sur le navigateur web du client, les différentes composantes avec lesquelles notre application doit interagir. Il s'agit d'une part du LMS, alimenté en données par les apprenants et dont ces données doivent être récupérées par notre application. Nous proposons ici de réaliser cela par un greffon logiciel situé sur le serveur du LMS, convertissant ces données dans un format d'export. Les données ainsi formatées seront ensuite transmises à notre application via Internet. Côté application aura lieu une nouvelle conversion des données permettant leur import dans la base de données. D'autre part figurent les sources des autres données. L'import de celles-ci étant manuel, nous n'avons pas rajouté de composant applicatif particulier pour elles, mais elles devront également passer par le module de conversion et d'import.

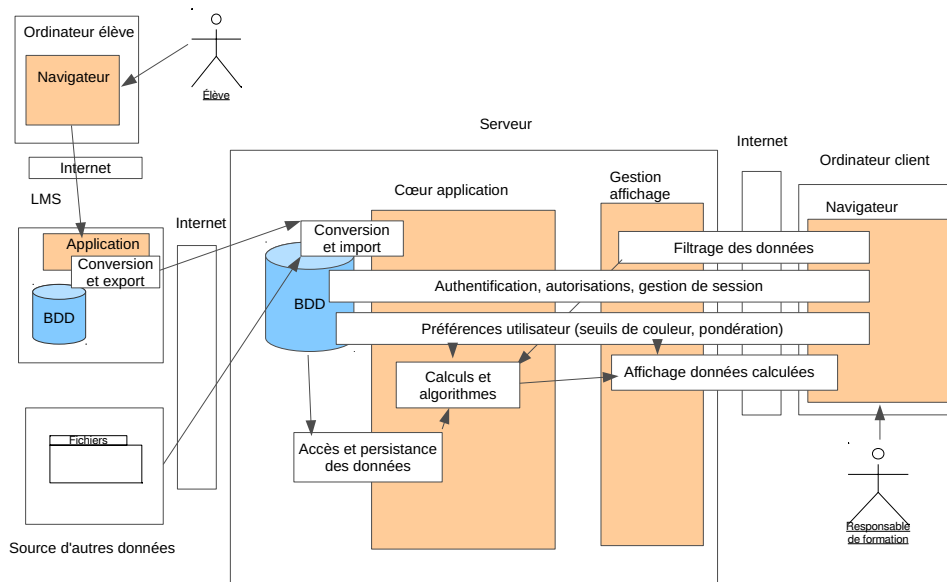


FIGURE 5.2 – Notre proposition d'architecture applicative pour GIGA

5.1.4 Architecture technique

Par rapport à l'architecture applicative, l'architecture technique fait des choix d'implémentation matériels et logiciels. Elle peut se préoccuper des aspects quantitatifs de dimensionnement et du déploiement. On peut y décider le regroupement de fonctionnalités en bibliothèques.

Il est classique d'avoir un découpage en 5 grandes parties, également nommées tiers ou niveaux :

- le niveau de présentation
- le niveau application
- le niveau « métier sans persistance », qui contient la représentation objet des objets processus et entités
- le niveau d'accès aux ressources, qui gère la correspondance entre les données sous leur forme stockée et leur forme manipulée dans l'application
- le niveau de ressources, qui s'occupe du stockage des données.

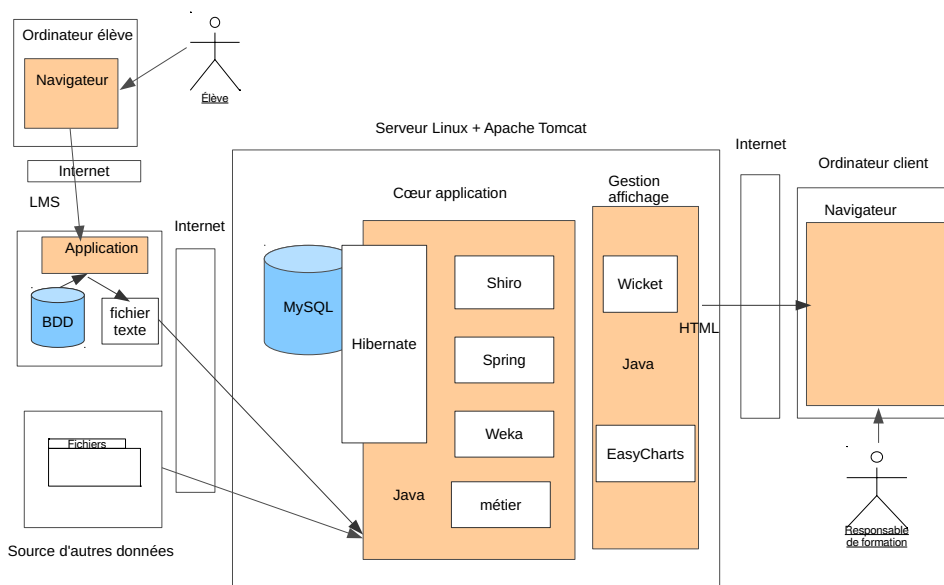


FIGURE 5.3 – Notre proposition d'architecture technique pour GIGA

Nous avons choisi d'utiliser côté serveur le langage Java avec les frameworks Hibernate, Spring, Shiro et Wicket pour l'application, et MySQL pour la base de données. Hibernate est un ORM (mapping objet relationnel) qui permet de faire le lien entre la base de données et un code orienté objet, en gérant notamment la persistance des objets. Shiro gère l'aspect authentification et gestion de session. Spring permet d'utiliser un type particulier d'objets, les « beans », qui peuvent être décrits en XML ou avec des annotations. Wicket gère l'aspect Web en fournissant des modèles objets correspondant aux éléments du site et en fournissant une correspondance entre objet et code HTML.

Le serveur utilisé est Apache Tomcat, installé sur Linux. Pour la partie apprentissage automatique, nous avons choisi d'utiliser la librairie open source Weka

[Machine Learning Group at the University of Waikato, 2013], qui fournit une API Java et qui est très réputée. La librairie EasyCharts nous fournira un moyen facile de faire des graphiques. À part Weka, que nous avons choisi, ces technologies étaient déjà utilisées par Andil pour ses autres produits, ce qui a mené à ce choix, et a permis de se concentrer sur le cœur de l'application.

Tous ces choix sont récapitulés dans le schéma 5.3.

5.1.5 Architecture orientée données

Nous avons défini un nouveau schéma dont le point de vue est technico-fonctionnel, mais qui est centré sur la circulation des données à travers notre application. Ce schéma est présenté dans la figure 5.4. Ce schéma se veut très récapitulatif de notre projet et présente donc un cadre un peu élargi par rapport à la simple application GIGA : on y voit aussi figurer nos propositions concernant la création de tuteur virtuel intelligent.

Comme on le voit, nos données sont créées en deux lieux : sur le LMS, par l'interaction de l'étudiant avec les contenus fournis par les tuteurs, et dans le système de données de la formation. Ces deux types de données suivent par la suite un même chemin, qui les conduit, par un import, à être centralisées et intégrées dans la base de données de GIGA. GIGA va ensuite manipuler ces données pour calculer nos indicateurs, puis ces indicateurs pour appliquer les méthodes d'apprentissage automatique que nous avons proposées. Enfin, toutes ces sorties peuvent être réutilisées pour servir d'entrées à un moteur comportemental permettant de générer les actions d'un tuteur virtuel intelligent.

5.2 Conception

Des détails liés à la conception de GIGA ainsi que divers diagrammes de classes sont présentés en annexe B. On y trouvera notamment la structure générale de GIGA, les entités liées aux indicateurs, et des détails pour l'interface avec Weka et avec notre implémentation de HTM-CLA. Nous souhaitons rajouter ici quelques détails liés à la structuration des données à présenter en entrée des algorithmes basés sur HTM-CLA.

Les données temporelles présentées en entrée de notre hiérarchie HTM-CLA sont tout naturellement les traces d'utilisation produites par les utilisateurs du LMS, puis stockées dans GIGA. Actuellement, la réception de ces logs dans GIGA ne se fait pas en temps réel, mais une fois par jour, lorsqu'un bot les exporte de Moodle puis les importe dans GIGA. Dès lors, l'aspect temporel des données ne sera pas tout à fait bien représenté, même si on fait attention à présenter ensuite les logs dans leur ordre d'arrivée. Pour pallier ce problème, nous avons décidé que l'horodate du log ferait partie des caractéristiques présentées en entrée du HTM. Nous espérons que cela aidera à clarifier les choses, sans introduire une confusion entre ce qui relève du domaine spatial et ce qui relève du temporel.¹

1. Dans [Numenta, 2013], Numenta décrit davantage son système et dit (traduction personnelle) :

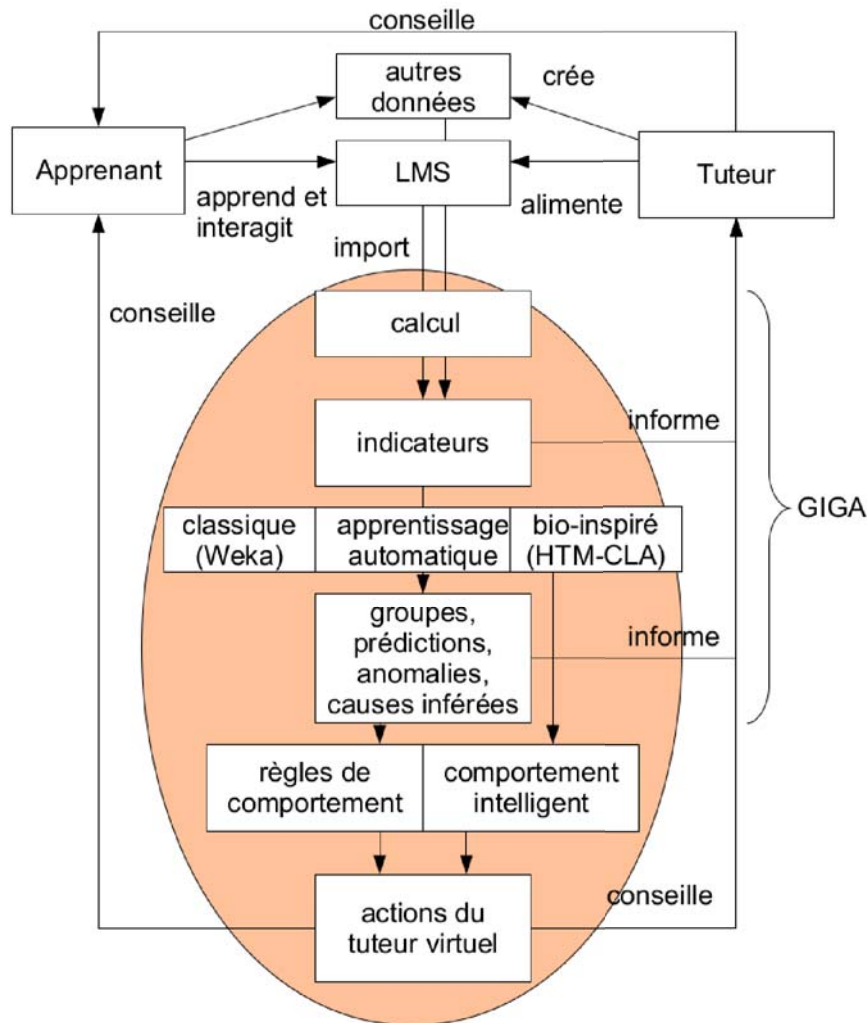


FIGURE 5.4 – Notre proposition d'architecture de données pour GIGA

Par la suite, on pourrait réfléchir à la pertinence d'un connecteur plus avancé qui envoie continuellement les nouveaux logs de Moodle vers GIGA, et dont la réception déclenche la mise à jour du HTM ; ou simplement de demander une actualisation plus fréquente. Nous aurions également l'avantage d'avoir des données plus à jour sur le reste de GIGA. Il faut toutefois éviter qu'un tel connecteur consomme trop de ressources de calcul, d'autant plus s'il déclenche de nouveaux calculs côté GIGA : actuellement, un cache est utilisé pour tous les calculs, en se basant sur le fait que

« Les valeurs des indicateurs sont envoyées automatiquement à Grok, typiquement toutes les 5 minutes. Elles sont combinées à un horodatage, qui aide Grok à apprendre les motifs reliés au moment de la journée ou au jour de la semaine. » Nous y trouvons donc une justification à notre approche d'horodatage.

les données ne changent que tous les jours, et que les calculs n'ont donc pas besoin d'être refaits plus fréquemment ; sans ce cache, le temps de chargement de certaines pages deviendrait bien plus important.

5.3 GIGA, une application dédiée au suivi des étudiants

5.3.1 But

GIGA a pour but de permettre aux responsables d'une formation en ligne mise en place sur une plate-forme classique ne comprenant qu'un LMS (Moodle, par exemple) de suivre les élèves inscrits à cette formation tout au long de leur parcours. GIGA ne nécessite aucune installation intrusive côté LMS pour pouvoir fonctionner. GIGA permet à la fois un suivi individualisé de chaque élève et un suivi collectif d'avancement de la promotion en cours. Ce suivi a notamment pour but de repérer plus facilement un élève en difficulté avant son décrochage complet et irrémédiable : ainsi, si un élève ne se connecte pas ou très rarement, cela sera signalé.

Pour l'instant, toute action est laissée à l'initiative du responsable de formation, afin de pouvoir prendre en compte ce qu'il peut savoir des situations individuelles des apprenants.

Le suivi collectif de l'ensemble des élèves d'une formation permet de déterminer le comportement global des élèves. Il peut permettre de voir quels contenus ont déjà été lus et assimilés, et si les outils de communication mis en place (forums...) sont bien employés. On peut ainsi voir si un recadrage est nécessaire pour l'ensemble de la promotion.

Par ailleurs, GIGA permet également la comparaison entre plusieurs promotions d'une même formation, afin de situer les apprenants en cours de formation à ceux dont on connaît déjà les résultats.

À terme, GIGA pourra également réutiliser les indicateurs d'un étudiant afin de prédire son parcours et le positionner parmi plusieurs catégories d'élèves que nous sommes en train de déterminer.

5.3.2 Technologie

Andil a fait le choix de présenter GIGA sous la forme d'une application web pour rendre ces données facilement accessibles à ses utilisateurs, à savoir les responsables de formations. Cette application utilise des technologies reconnues, qui ont été présentées dans la partie 5.1.4.

5.3.3 Données traitées

GIGA se base sur des données issues d'un LMS où travaillent des apprenants. Sans modifier en rien ce LMS, GIGA en aspire certaines données pertinentes, notamment les traces d'utilisation. On peut toutefois enrichir les informations fournies par le LMS pour enrichir les traitements réalisés par GIGA.

Actuellement, seul le connecteur entre GIGA et Moodle a été réalisé, mais il est facilement adaptable à d'autres LMS qui emploieraient un système semblable d'enregistrement de traces d'utilisation.

Les traces d'utilisation fournies par le LMS sont limitantes en terme de données récupérables et utilisables pour GIGA. Ainsi, par exemple, sur Moodle, on peut savoir chaque fois qu'un apprenant est entré dans une leçon pour la lire, mais on ne peut pas connaître avec précision les durées des sessions. En conséquence, Andil a également réalisé un plugin pour Moodle permettant de récupérer ces données supplémentaires, de façon à pouvoir calculer le temps passé en ligne par chaque utilisateur.

Le choix exact des données récupérées par GIGA a été présenté dans la partie 2.1.

5.3.4 Fonctionnalités

Les principales fonctionnalités de GIGA se résument en trois grands thèmes :

- centralisation de données liées à une formation issues de plusieurs sources
- automatisation du travail du responsable de formation / de suivi
- examen de données brutes et traitées, à différents niveaux de granularité

5.3.4.1 Centralisation de données

GIGA permet de regrouper les données issues des logs de LMS et les autres données dans une seule base de données. Notre application de suivi d'élèves et d'amélioration de formation devra idéalement n'utiliser que cette base comme source de données. Nous proposons en conséquence des méthodes d'import de données existantes et de rajout manuel de données.

Pour l'instant, seul l'import automatique des logs de Moodle est réalisé, mais notre architecture de données est suffisamment générique pour pouvoir être élargie rapidement à l'avenir à d'autres LMS comme sources de données. Cet import est déclenché automatiquement tous les jours par deux scripts, un côté LMS pour exporter les nouvelles données et un côté GIGA pour les récupérer. La fréquence d'import peut être modifiée à la demande.

Par ailleurs, il est également possible de rentrer manuellement dans l'application, par des formulaires adaptés, les notes obtenues hors Moodle (par exemple aux examens finaux), la présence aux présentiels ainsi que des notes prises sur un élève. Il pourra être défini plus tard diverses méthodes de standardisation du format de ces données permettant un import automatique.

5.3.4.2 Automatisation du travail du responsable de formation

L'import automatique présenté plus haut se charge de récupérer les données de Moodle. Pour les autres données, pour l'instant l'import est manuel, mais cela pourrait changer prochainement, notamment pour les notes finales et les présentiels.

Certaines formations réclament un certificat prouvant que l'élève a passé un certain nombre d'heures connecté sur la plate-forme à réaliser des activités. GIGA est également capable d'éditer automatiquement ce type de certificat.

5.3.4.3 Présentation de données brutes et traitées

Nous avons défini précédemment dans le chapitre 2 les différentes données brutes, statistiques et indicateurs qui nous semblaient pertinents au suivi d'un élève. Toutes ces données seront calculées et affichées dans GIGA.

Nous avons également défini dans la partie 3 les utilités que nous voyions aux méthodes d'apprentissage automatique et les diverses sorties que nous souhaitions obtenir grâce à ces méthodes. Nous avons implémenté ces fonctionnalités dans GIGA, où elles sont également calculées et affichées.

Les logs fournis par le LMS sont limitants en terme de données récupérables et utilisables pour GIGA. Ainsi, par exemple, sur Moodle, on peut savoir chaque fois qu'un apprenant est entré dans une leçon pour la lire, mais on ne peut pas connaître avec précision les durées des sessions. En conséquence, Andil a également réalisé un plugin pour Moodle envoyant régulièrement un signal au serveur pour signaler que l'utilisateur est encore connecté, de façon à pouvoir calculer le temps passé en ligne par chaque utilisateur de façon plus fiable. Le temps passé en ligne ne pourra être considéré comme fiable qu'à partir du déploiement de ce plugin sur le LMS.

Les indicateurs que nous avons choisis sont basés sur plusieurs critères : les données disponibles dans Moodle et ailleurs, la facilité de compréhension mais aussi la couverture la plus complète possible de l'intégralité des activités d'un apprenant. Ils reprennent souvent nos statistiques et les convertissent en une note.

La pondération entre indicateurs est personnalisable par l'utilisateur de GIGA. Cela lui permet de faire en sorte que la note globale reflète au mieux ce qu'il juge important.

Apprentissage automatique Nous proposons dans notre application d'utiliser les méthodes suivantes, fournies par Weka :

- pour le partitionnement de données : Expectation Maximisation, Hierarchical Clustering, Simple K-Means, and X-Means.
- pour la classification : BayesNet, ClassificationViaClustering, MultilayerPerceptron, NaiveBayes, SimpleLogistic, SimpleLinearRegression, DecisionTable, Logistic, LinearRegression, ID3, M5P
- pour la régression : MultilayerPerceptron, SimpleLogistic, SimpleLinearRegression, Logistic, LinearRegression

Visualisation des données Nous avons besoin que ces indications soient très visuelles, avec des graphiques pertinents proposés et une colorimétrie permettant de détecter au premier coup d'œil si un élève est en difficulté par rapport à ses collègues. Ainsi, GIGA affiche les données sous forme de tableaux avec des codes couleurs aisément compréhensibles (rouge, jaune, vert) permettant de les rendre claires et lisibles, ainsi que des graphiques permettant de visualiser ces données de façon plus intuitive et immédiate. Beaucoup de ces données sont immédiatement reliées par un lien hypertexte à une vue plus détaillée permettant de comprendre comment l'information a été calculée.

On peut sélectionner une période, une formation, un cours, voire même un groupe d'élèves grâce au filtre proposé sur chaque page.

Les seuils permettant de définir la couleur associée à la note sont personnalisables. Cela reflète le fait que nos indicateurs sont relatifs et non absolus, et que les formations peuvent être très différentes entre elles. En conséquence, les valeurs acceptables peuvent varier grandement d'une formation à l'autre.

Interface L'interface de GIGA est divisée en deux grandes parties, l'une pour les données individuelles des étudiants, une autre pour les formations entières. Au sein de chacune de ces parties, on retrouve les même données, mais présentées sous forme de moyenne pour les formations.

Dans les images suivantes, les noms ont été floutés afin de préserver l'anonymat des apprenants. La figure 5.5 présente la page d'accueil de GIGA, et la figure 5.6 en présente la page regroupant les différents indicateurs. D'autres captures d'écran de GIGA sont disponibles en annexe A.



FIGURE 5.5 – La page d'accueil de GIGA, centrée sur une formation dont on voit l'IGA

5.3.5 Comparaison

Un système similaire de remontée d'information aux tuteurs est présenté dans [Feng and Heffernan, 2005], mais il est basé sur un tuteur virtuel ad hoc, contrairement à notre application qui est générique, pouvant s'adapter à de nombreux LMS. Il en va de même pour TADA-Ed, proposé dans [Merceron et al., 2004], qui a en commun avec GIGA l'utilisation de méthodes d'apprentissage automatique.

GISMO (Graphical Interactive Student Monitoring System for Moodle) [Mazza and Milani, 2005] est une applet Java pour Moodle qui permet de visualiser l'activité des élèves en fonction du temps. Il permet d'avoir une vue d'ensemble pour tout le groupe. En

Etudiants	Lecture de leçons	Lecture de PDFs	Consultation de ressources	Fréquence de connexion	Dernière connexion	Quiz effectués	SCORMs effectués	Devoirs effectués	Hot Potatoes effectués	Notes obtenues	Dernières notes obtenues	Meilleures notes obtenues	Lecture du forum	Nouveaux sujets sur le forum	Réponses sur le forum
	2,43	10	4,06	2,85	9,63	0	2,37	0	4,86	5,35	4,62	5,61	2,55	0	0
	0,13	0,12	0,41	0,19	3,08	0	0,2	0	1,53	0	0	0	0,2	0	0
	1,12	7,05	0,49	1,99	9,68	0	0	0	1,53	0	0	0	1,7	2,61	0
	5,79	7,78	3,76	4,95	9,68	0	8,41	4,07	8,89	7,53	9,21	9,16	4,35	0,43	0
	0,26	1,84	1,77	0,81	2,83	0	1,74	1,37	7,5	0	0	0	0,13	0	0
	3,13	8,17	10	7,12	10	0	5,02	0,44	7,92	6,13	0,86	6,43	1,92	1,74	0
	0,13	0,12	0,35	0,08	1,99	0	0	0	0	0	0	0	0,14	0	0
	7,33	8,03	2,04	3,09	7,73	0	4,5	1,45	4,72	4,57	2,66	6,21	1,46	1,3	0
	0,72	7,87	2,66	1,51	9,32	0	8,67	0	8,89	8,08	9,33	9,52	2,06	0	0
	10	9,59	5,06	10	10	0	9,43	7,06	10	7,09	8,82	8,83	10	10	0
	3,57	8,22	5,49	3,92	8,88	0	3,59	0	6,25	3,52	2,05	5,05	3,53	3,91	0

FIGURE 5.6 – Les différents indicateurs et leur valeur pour tous les étudiants de la formation sélectionnée

revanche, il n'agrège pas les données en fonction du temps. L'interface est présentée en figure 5.7².

MooDog (Moodle Watchdog) [Zhang et al., 2007] est un greffon pour Moodle qui permet de visualiser dans Moodle des traces des utilisateurs. Il fournit des données agrégées en statistiques, avec visualisation associée, pour le groupe et pour l'élève, et identifie en outre les activités qu'un étudiant n'a pas encore faites, allant jusqu'à rappeler aux élèves d'aller regarder des contenus pédagogiques. Toutefois, il est uniquement capable de fonctionner sur Moodle et ne recherche pas la généricité. L'interface est présentée en figure 5.8³.

ProgressTracker [Vermeulen et al., 2014] est également un greffon pour Moodle, mais cette fois à destination des apprenants eux-mêmes. Les données d'un élève sont agrégées en statistiques qui permettent de positionner l'étudiant vis-à-vis du groupe. L'étudiant peut également se fixer des buts. Une proposition d'interface est présentée en figure 5.9⁴. Là encore, cet outil est uniquement conçu pour Moodle. Ils n'ont pas non plus cherché à distinguer lecture et écriture sur le forum.

Vermeulen a récolté des données objectives et des réponses à un questionnaire sur l'utilisation de ProgressTracker par les étudiants. Il en est ressorti que les étudiants qui avaient utilisé ProgressTracker pour se fixer un but utilisaient davantage la plate-forme et débutaient leurs devoirs plus en avance. Toutefois, l'auteur souligne

2. Source : [Mazza and Milani, 2005]

3. Source : [Zhang et al., 2007]

4. Source : [Vermeulen et al., 2014]

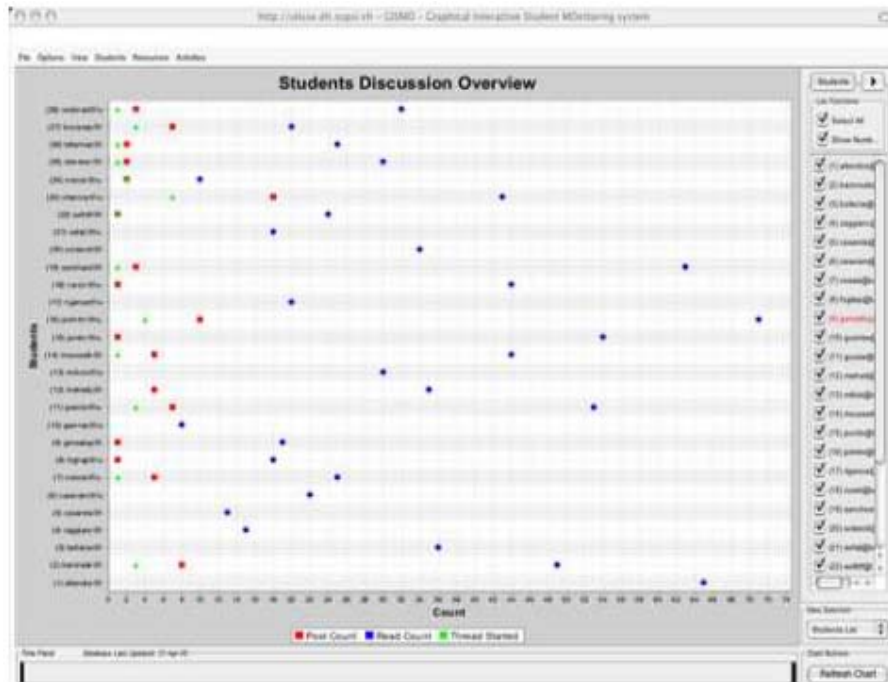


FIGURE 5.7 – L'interface de GISMO montre l'activité des élèves en fonction du temps

qu'il est impossible de déterminer si c'est l'utilisation de ProgressTracker qui les engage davantage dans le processus d'apprentissage, ou s'ils l'utilisent parce qu'ils sont plus engagés. Par ailleurs, les réponses à son questionnaire étaient très positives, et sont présentées dans la figure 5.10⁵.

L'application GIGA est déjà commercialisée par la société Andil auprès de plusieurs clients, dont les retours sont positifs.

5.4 Tests et résultats

Nous avons pu mener de premières expériences de partitionnement de données sur des données issues de formations réelles et importées dans GIGA, grâce à notre institut de formation partenaire, Juriscampus. Notre objectif via ces expériences est de tester notre proposition d'utilisation du partitionnement de données : peut-on l'utiliser pour repérer des groupes d'élèves de comportement différent ? Combien de profils d'élèves différents trouve-t-on ? Les différences entre les groupes sont-elles plutôt de nature quantitative ou qualitative ?

Avant toutefois de pouvoir mener ces expériences, nous avons dû nous définir un

5. Source : [Vermeulen et al., 2014]



Figure 3: Per-user activity as measured by Moodog.

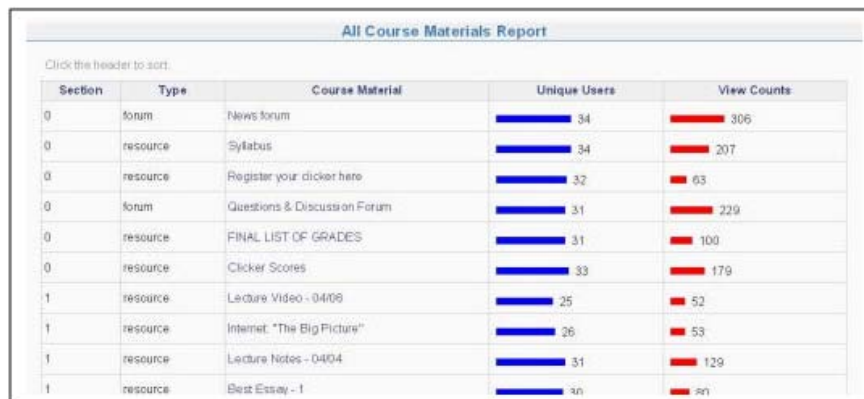


Figure 4: Resource access coverage as measured by Moodog.

FIGURE 5.8 – L'interface de MooDog fournit agrégation de données et visualisation

protocole. Avant de le présenter, nous allons expliquer les enjeux qui ont conduit à sa définition.

5.4.1 Considérations de protocole

Les méthodes d'exploration de données et d'apprentissage automatique partagent quelques grands concepts qui sont récapitulés ici. Ces concepts permettent de mesurer l'erreur commise par un algorithme et de s'assurer de ne pas tomber dans certains pièges.

Dimensions Le premier, Richard Bellman [Bellman, 1957] a mentionné que la grande dimensionalité des données était un problème fondamental en intelligence artificielle. Dans le domaine de l'apprentissage, en particulier, la complexité croît en exponentielle du nombre de dimensions des données, car c'est ainsi que croît l'espace des hypothèses possibles à explorer. C'est ce qu'il a appelé la malédiction de

152 Chapitre 5. Proposition d'application dédiée au suivi des étudiants

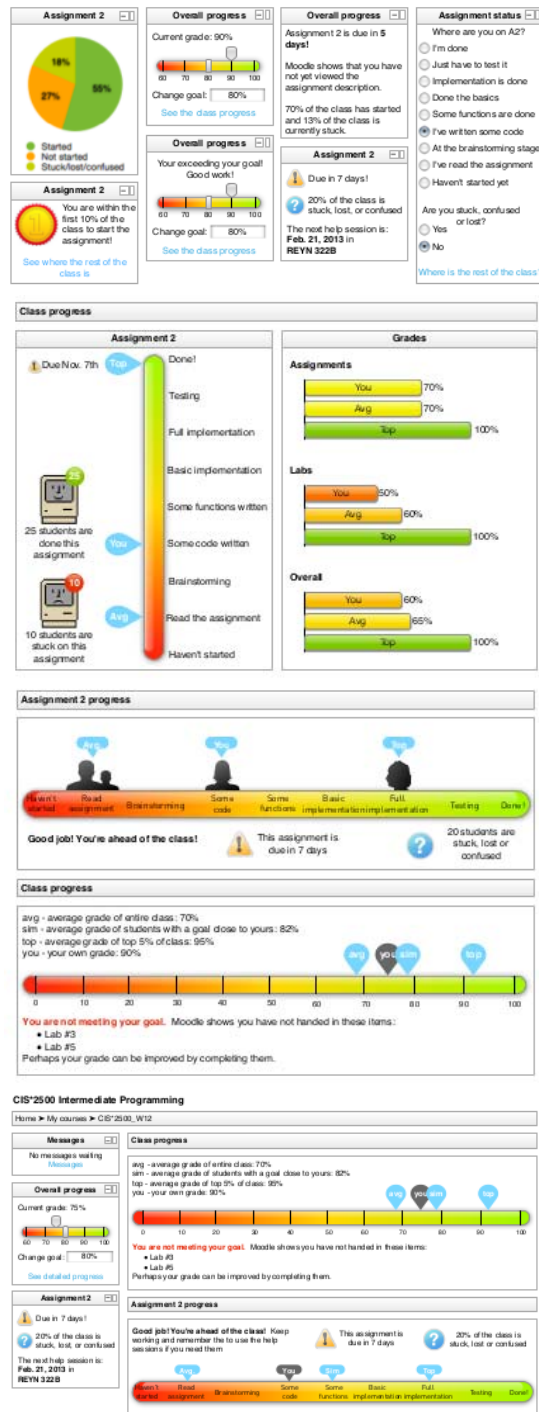


FIGURE 5.9 – L'interface de ProgressTracker fournit agrégation de données et visualisation

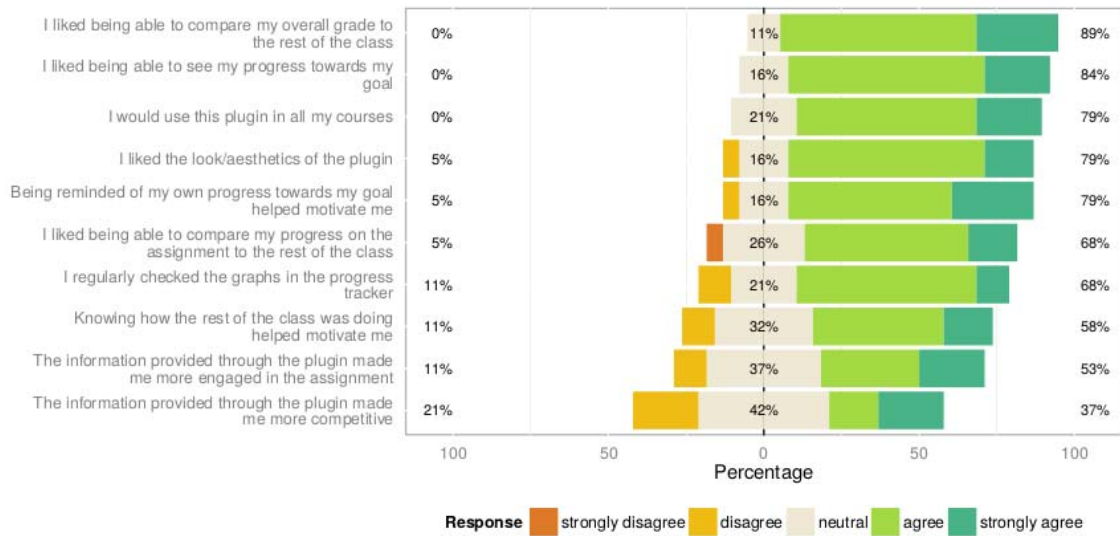


FIGURE 5.10 – Les questionnaires de satisfaction des étudiants utilisant Progress-Tracker

la dimensionalité.

Cette malédiction engendre deux conséquences : la première est que la quantité de données nécessaire pour parvenir à la convergence du modèle croît également de façon exponentielle, et la deuxième est que la durée du calcul augmente encore davantage, puisqu'il lui faut parcourir davantage de données et un espace d'hypothèses plus grand. On voit donc qu'un grand nombre de dimensions pose un problème majeur à l'exploration de données.

Pour venir à bout de ce problème, la manière la plus fréquemment employée est justement de réduire ce nombre de dimensions dès lors qu'elles dépassent le nombre de 10, par un pré-traitement des données. On nomme extraction de caractéristiques [Duda et al., 2000] le pré-traitement qui consiste à employer un algorithme pour transcrire les données dans un nouveau jeu de dimensions, dont le nombre pourra être réduit avec un minimum de perte d'information. Le principal algorithme utilisé pour cela est l'analyse en composantes principales, qui transforme les variables en les décorrélant. Les nouvelles variables, ou « axes principaux », sont classées par ordre d'importance, on peut donc ne garder que les n premiers. On peut aussi faire une sélection des dimensions originelles les plus pertinentes, c'est-à-dire qui contiennent le moins de redondances. Après cela, les traitements habituels, par exemple la classification, peuvent se dérouler normalement. On peut donc juger, comme Arel [Arel et al., 2010], que l'intelligence des traitements est répartie entre l'extraction de caractéristiques et l'algorithme de traitement à proprement parler.

Compromis sous/sur-apprentissage Tout apprentissage doit faire l'objet d'un compromis délicat pour éviter le sur-apprentissage. Ce problème arrive quand le

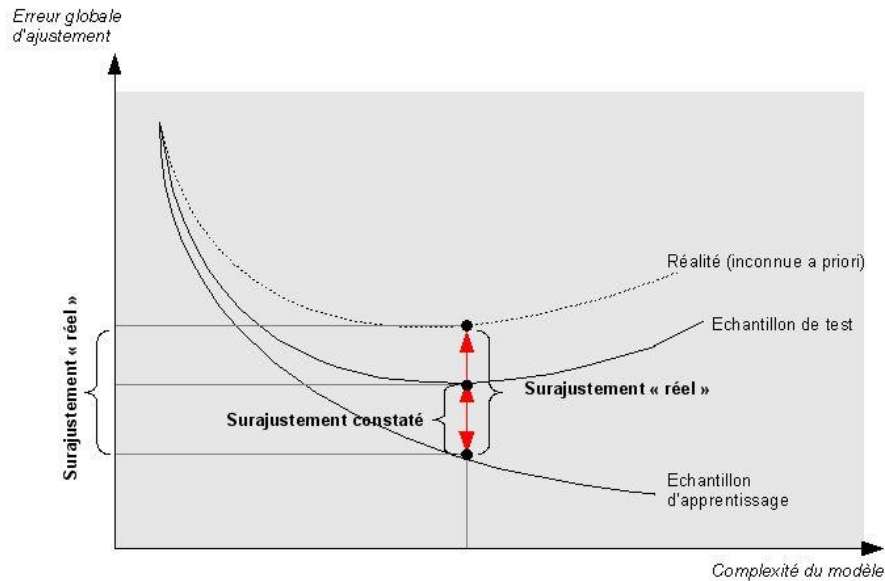


FIGURE 5.11 – Le phénomène du sur-apprentissage

modèle s'ajuste trop bien à tous les échantillons utilisés pour l'apprentissage, mais perd en pouvoir de généralisation à de nouvelles données. Il est illustré à la figure 5.11. Ce problème peut venir d'un modèle trop complexe, ayant une trop grande capacité de stockage d'information associé à un trop grand nombre d'échantillons présentés : puisqu'il peut reproduire exactement leur comportement, c'est ce qu'il fait, incluant donc les données bruitées, ou tout simplement le manque de représentativité complète inhérent à l'échantillonnage.

Il faut donc faire un compromis entre la bonne performance du modèle et son pouvoir de généralisation. Un modèle complexe offrira des meilleurs résultats, mais aura des risques de tomber dans le sur-apprentissage. On parle également de « dilemme biais-variance » : un modèle très simple offrira une représentation biaisée des données, c'est-à-dire avec une erreur incompressible, mais cette représentation variera moins en fonction de l'échantillon sélectionné, et inversement pour un modèle plus complexe.

Pour éviter cette erreur, il faut toujours laisser de côté un certain nombre de données, qui serviront à effectuer une validation du modèle. Si l'on trace les courbes de l'erreur du modèle sur l'échantillon d'apprentissage et sur l'échantillon de validation, ces deux courbes décroissent initialement mais alors que l'erreur sur l'échantillon d'apprentissage continue à décroître, celle sur l'échantillon de validation finit par augmenter à nouveau, signalant de façon graphique le sur-apprentissage. On souhaite donc se placer au point qui minimise l'erreur sur l'échantillon de validation, pour avoir le modèle le plus précis possible, sans sur-apprentissage.

Une autre façon d'éviter le sur-apprentissage est d'introduire ce qu'on appelle une régularisation, qui consiste en l'application d'une pénalité sur les modèles trop

complexes, qui permet, à performance presque égale par ailleurs, de favoriser le modèle le plus simple. On voit directement le lien avec le rasoir d'Occam. Bien souvent, les modèles complexes ont des paramètres aux valeurs assez extrêmes, ce qui permet un algorithme simple de régularisation employant une norme sur les valeurs des différents paramètres.

Équivalence bayésienne et théorie de l'information En 1946, le théorème de Cox [Cox, 1946] permet pour la première fois d'utiliser le calcul de probabilités pour modéliser de façon numérique une notion de plausibilité. Ce théorème ouvre la porte vers l'interprétation logique, ou bayésienne, et non fréquentielle des probabilités, telle que suggérée par Myron Tribus [Tribus, 1969], où elles représentent un état de connaissance. Jaynes [Jaynes, 1994] reprend ce théorème pour montrer que tout apprentissage de modèle par induction, c'est-à-dire par exemples successifs, est équivalent, à un homomorphisme près, à un apprentissage par inférence bayésienne. Un apprentissage qui ne vérifierait pas cette propriété conduirait nécessairement à des incohérences. La convergence de ces méthodes a d'ailleurs été démontrée par Tribus.

Il en découle également que tout modèle d'apprentissage doit nécessairement choisir la distribution de probabilité maximisant l'entropie, c'est-à-dire qui rajoute le moins possible d'information par rapport aux observations. Pour comprendre cela, nous devons faire une petite incursion par le domaine de la théorie de l'information.

Le rasoir d'Occam ou d'Occam est un principe de raisonnement philosophique nommé d'après le philosophe franciscain du XIVe siècle Guillaume d'Ockham, mais connu au moins depuis Aristote. On le trouve également appelé principe de simplicité, principe d'économie ou principe de parcimonie. Il peut se formuler comme suit : « les hypothèses suffisantes les plus simples sont les plus vraisemblables ». C'est un des principes heuristiques fondamentaux en science, sans être pour autant à proprement parler un résultat scientifique.

Ce principe a été formalisé sous forme mathématique par l'induction de Solomonoff [Solomonoff, 1964]. Sous l'hypothèse que l'environnement suit une loi de probabilité inconnue mais calculable, on peut formaliser les données comme ayant été générées par un algorithme inconnu. L'ensemble des hypothèses possibles pour expliquer les données est donc l'ensemble des entrées possibles d'une machine de Turing universelle. L'algorithme d'induction proposé consiste donc à examiner toutes les hypothèses de cet ensemble une à une, en vérifiant leur correspondance avec les données. On accorde ensuite à toutes les hypothèses qui correspondent une probabilité relative proportionnelle à leur parcimonie, formalisée comme le nombre de bits de leur représentation binaire :

$$P = \frac{1}{2^n}$$

Ces probabilités sont ensuite normalisées au fur et à mesure, pour que la somme des probabilités de toutes les hypothèses valables vaille bien 1.

Cette méthode donne un résultat si précis et si complet qu'il a parfois été surnommé « le sabre laser de Solomonoff ». Toutefois, cet algorithme ne converge

pas en un temps fini, et n'est donc pas utilisable tel quel.

On retrouve des notions équivalentes dans le domaine de la théorie de l'information, qui dispose d'une notion nommée l'entropie d'une information, ou entropie de Shannon [Shannon, 1948]. Cette grandeur représente la taille minimale de cette information après une compression sans perte, donc son contenu réel en termes d'information non redondante. Cela correspond aussi à l'imprédictibilité de cette information.

On la mesure formellement par la formule suivante :

$$H(X) = E(-\log(P(X))) = E(I(X))$$

Avec H l'entropie de Shannon, E l'espérance et I l'auto-information de X , ce qui correspond aussi au degré de surprise qu'on peut ressentir à la vue du résultat [Tribus, 1961].

En 1957, Edwin Thompson Jaynes démontra [Jaynes, 1957] le lien formel existant entre l'entropie thermodynamique macroscopique (elle-même équivalente à l'entropie microscopique de Boltzmann), qui est une mesure du désordre, et l'entropie mathématique de Shannon.

En conséquence, l'entropie peut nous permettre de distinguer, entre deux hypothèses probables pour expliquer un phénomène donné, celle qui paraît la moins arbitraire. Il s'agit, d'après le rasoir d'Occam, de celle qui contient le moins d'information superflue, et donc de celle qui a la plus grande entropie au sens de Shannon. C'est ce qu'on appelle le principe d'entropie maximale.

L'ouvrage de Tribus [Tribus, 1961] popularise la notation des probabilités en décibels, qui rend la règle de Bayes additive et permet de quantifier de façon unique l'apport d'une observation en la rendant désormais indépendante des diverses estimations a priori préalables.

La vraisemblance On utilise également couramment en statistique une grandeur appelée la vraisemblance, qui est en fait une fonction de probabilité conditionnelle, visant à estimer la probabilité d'une observation donnée sachant un certain nombre de paramètres. Par exemple, si une pièce est tombée 10 fois de suite sur son côté pile, on peut s'interroger sur la vraisemblance qu'elle ne soit pas pipée. On peut donc s'en servir à l'envers : estimer l'endroit où cette fonction atteint son maximum permet de déterminer les paramètres les plus probables permettant d'expliquer une observation donnée. Cette méthode s'appelle, en toute logique, l'estimation du maximum de vraisemblance.

Pour de nombreux usages, il est plus pratique de travailler avec le logarithme de cette fonction de vraisemblance, qu'on appelle donc log-vraisemblance (*log-likelihood* en anglais). Cela ne change pas la valeur du maximum, puisque le logarithme est à croissance monotone, et facilite souvent les calculs formels de dérivées, en changeant les produits de facteurs en sommes. Cela permet également d'examiner indépendamment l'apport des différents facteurs.

Évaluation de la qualité d'un modèle Afin d'avoir les meilleurs résultats possibles, et pour éviter de tomber dans les travers listés plus haut, diverses méthodes permettent d'évaluer la qualité des résultats obtenus. Il est indispensable de les utiliser afin de s'assurer d'avoir des résultats pertinents.

Outils statistiques Certains outils, issus des statistiques, permettent de vérifier la performance des méthodes de type classification de données. Ce sont :

- le taux d'erreur de classement
- la matrice de confusion
- la courbe ROC
- la courbe de lift

Ces méthodes se basent sur le comptage des données respectivement bien et mal classifiées en deux catégories appelées « positifs » et « négatifs », qui permet de définir quatre catégories : les vrais positifs et vrais négatifs, qui sont respectivement bien classifiés dans leur catégorie réelle d'appartenance ; et les faux positifs et faux négatifs, qui sont classifiés dans la mauvaise catégorie.

À partir de ce comptage, on peut définir d'autres grandeurs pertinentes :

- la sensibilité, qui correspond au taux de vrais positifs : $\alpha(s) = \frac{\#VP}{P}$
- la spécificité, qui correspond au taux de vrais négatifs : $\beta(s) = \frac{\#FN}{N}$

La courbe Receiver Operating Characteristic (ROC) consiste à tracer la sensibilité en fonction de $1 - \beta(s)$. La courbe de lift consiste à tracer la sensibilité en fonction du pourcentage d'individus en-dessous du seuil.

Critères d'information À partir des principes fortement reliés de vraisemblance, de théorie de l'information et d'entropie dont nous avons parlé plus haut, plusieurs critères ont été proposés pour mesurer objectivement la qualité d'un modèle donné. Il est à noter que ces critères incluent souvent une forme de régularisation.

Le critère d'information d'Akaike [Akaike, 1974], (en anglais *Akaike information criterion* ou AIC) a été proposé par Hirotugu Akaike en 1973. Il se base sur le calcul de la log-vraisemblance, mais ajoute une pénalisation pour le nombre de paramètres du modèle, afin de se rapprocher du critère de parcimonie. On choisit alors le modèle avec le critère d'information d'Akaike le plus faible.

Le critère d'information d'Akaike s'écrit comme suit :

$$AIC = 2k - 2 \ln(L)$$

avec k le nombre de paramètres du modèle et L la valeur maximale de log-vraisemblance du modèle. Toutefois, ce critère ne permet que de comparer plusieurs modèles entre eux, et n'offre aucune mesure absolue de la qualité du modèle.

Le critère d'information bayésien [Schwarz et al., 1978] (en anglais *Bayesian information criterion* ou BIC) est un dérivé du critère d'information d'Akaike. Il a été proposé par Gideon Schwarz en 1978. Il s'en différencie par une pénalité qui dépend également de la taille de l'échantillon et pas seulement du nombre de

paramètres, ceci afin d'éviter le sur-apprentissage.

$$BIC = k \ln(N) - 2 \ln(L)$$

avec L la vraisemblance du modèle estimée, N le nombre d'observations dans l'échantillon et k le nombre de paramètres du modèle.

Le critère de la déviance de l'information (en anglais *Deviance information criterion* ou DIC) est une généralisation des critères précédents adaptées aux modèles hiérarchiques. Il est utile quand on cherche à traiter des données générées par une simulation de Monte Carlo utilisant une chaîne de Markov (MCMC en anglais), permettant alors un calcul direct ne requérant pas de passer par l'étape du maximum de vraisemblance.

Il introduit une grandeur, la déviance, définie comme $D(\theta) = -2 \log(p(y|\theta))$, où y est les données observables et θ est les paramètres inconnus du modèle. À partir de là, on peut calculer de deux sortes le nombre effectif de paramètres du modèle, p .

La première formule, proposée par Spiegelhalter et al. [Spiegelhalter et al., 2002] est $p = \bar{D} - D(\bar{\theta})$, où $\bar{\theta}$ est l'espérance de θ , et \bar{D} est l'espérance de D (donc plus elle est grande, moins le modèle est bon).

La seconde, proposée par Gelman et al. [Gelman et al., 2004] est $p = \frac{1}{2} \widehat{\text{var}}(D(\theta))$.

Pour finir, le critère de déviance de l'information est défini par la formule suivante :

$$DIC = p + \bar{D}$$

ou, de façon équivalente, et qui montre mieux le parallèle avec le critère d'Akaike :

$$DIC = D(\bar{\theta}) + 2p$$

Il s'agit également d'une valeur à minimiser : on cherche à minimiser à la fois la déviance et le nombre de paramètres du modèle.

Le critère de focalisation de l'information [Claeskens and Hjort, 2003] (en anglais *Focused information criterion*, ou FIC) ne s'intéresse pas à la qualité globale du modèle, mais à un seul paramètre considéré comme le plus intéressant, qu'on appellera μ , et dont on cherche à estimer la valeur en utilisant plusieurs modèles qui en fournissent chacun une estimation, qu'on notera μ_j pour le modèle j . On cherche donc à estimer la qualité de chaque estimateur. Il n'existe pas de formule universelle pour ce critère, puisque son calcul dépend du paramètre retenu et des mesures choisies pour la précision de chaque modèle.

Le critère d'information de Hannan-Quinn [Hannan and Quinn, 1979] (en anglais *Hannan-Quinn information criterion* ou HQC) est encore une autre variante, plus proche du critère bayésien, et qui utilise la formule suivante :

$$HQC = n \log \left(\frac{RSS}{n} \right) - 2k \log(\log(n))$$

où k est le nombre de paramètres du modèle, n le nombre d'observations, et RSS la somme des carrés de l'erreur. Il est en pratique très peu utilisé. [Burnham and Anderson, 2002]

Validation Comme nous l'avons mentionné au sujet du sur-apprentissage, il est toujours pertinent de vérifier que les structures trouvées dans les données d'apprentissage sont également présentes dans l'intégralité des données, surtout quand on utilise des méthodes à but prédictif. Cette phase de validation, également appelée validation croisée, utilise donc un jeu de données dit de test, qui n'était pas dans les données d'apprentissage. On observe si les structures détectées s'appliquent bien à ce jeu de données de test. Par exemple, si l'on a trouvé des règles permettant de déterminer si un e-mail est du spam, on vérifie si les e-mails de tests ont été correctement classifiés. L'inconvénient de cette méthode est qu'elle nécessite un étiquetage des données ou une relecture par un humain afin de procéder à cette vérification.

Il existe trois grandes techniques de validation croisée [Kohavi et al., 1995].

La première s'intitule la validation sur ensemble de test, également appelée validation par retenue. On pratique un échantillonnage aléatoire en divisant les données en un échantillon d'apprentissage ($> 60\%$ de l'échantillon) et un échantillon de test. On construit le modèle sur l'échantillon d'apprentissage, puis on le valide sur l'échantillon de test, ce qui nous fournit une indication d'erreur de prédiction. Cette erreur peut être calculée par diverses méthodes, par exemple l'erreur quadratique moyenne. Cette méthode a pour inconvénient de théoriquement permettre que certaines données importantes ne soient jamais sélectionnées parmi l'échantillon de test.

La deuxième, ou « validation croisée n fois », consiste à diviser les données en n échantillons. Ensuite, un des échantillons est sélectionné comme échantillon de test, et les autres constitueront l'échantillon d'apprentissage. On procède alors au calcul de l'erreur comme dans la validation sur ensemble de test, puis on répète l'opération en changeant l'échantillon de validation, jusqu'à avoir passé les n échantillons en revue. On calcule alors la moyenne de l'erreur obtenue. Cela permet d'utiliser toutes les données au moins une fois pour la validation, et à proportion égale pour l'apprentissage. On voit que si $n = 3$, on se rapproche de la méthode précédente, mais avec un moyennage de l'erreur.

La troisième, ou « validation croisée moins un », en anglais « leave-one-out cross-validation » (LOOCV), consiste à enlever une seule donnée en tant qu'ensemble de validation, et à répéter cette opération avec l'ensemble des données. C'est donc la même méthode que la validation croisée n fois, si l'on prend n égal au nombre de données.

Pour chacune de ces méthodes, elle peut être répétée un certain nombre de fois, et l'erreur moyennée, pour avoir une vision encore plus précise qui élimine le biais d'échantillonnage, ainsi que l'éventuelle construction non déterministe du modèle.

À l'issue de cette phase, si l'erreur est jugée trop importante, c'est-à-dire si les structures apprises ne se retrouvent pas suffisamment bien dans les données de test, il est nécessaire de revenir en arrière et de modifier le processus. Cela peut nécessiter de changer les pré-traitements ou les méthodes d'exploration de données. Sinon, si les structures sont validées à cette étape, on peut passer à la phase d'interprétation des résultats.

5.4.2 Protocole expérimental retenu

Dans un premier temps, nous avons converti nos données sous la forme d'une liste de notes individuelles correspondant aux indicateurs normalisés définis dans la partie 2.2. Nous avons ensuite transformé ces données en attributs et instances au format Weka. Les algorithmes de partitionnement de données de Weka prennent ces instances en entrée et renvoient en sortie des groupes de ces instances, que nous convertissons ensuite en groupes d'élèves. Ensuite, pour analyser sémantiquement ces groupes, nous examinons les indicateurs pour chaque groupe d'élèves. Le nombre de dimensions nous a paru suffisamment faible pour ne pas nécessiter de chercher à le réduire.

Pour tester la justesse de nos groupes, nous utilisons la validation croisée 10 fois, afin de vérifier si les résultats trouvés sur l'ensemble d'apprentissage sont généralisables à l'ensemble des données. Nous avons également pris une moyenne des résultats obtenus sur plusieurs simulations avec des graines aléatoires différentes.

Nous avons choisi les algorithmes suivants proposés par Weka: Expectation Maximisation, Hierarchical Clustering, Simple K-Means et X-Means. Parmi ces algorithmes, X-Means est à mettre à part puisqu'il détermine de lui-même le nombre de groupes qui lui semble le plus pertinent. Les trois autres prennent en paramètre le nombre de groupes souhaité. En conséquence, pour ceux-là (K-Means, EM et Hierarchical Clustering), nous avons fait plusieurs simulations en faisant varier ce paramètre, afin de voir quelle valeur donnait les résultats correspondant le mieux à la structure naturelle des données. Au vu de nos données (faible nombre d'élèves), nous avons choisi des valeurs allant de 2 à 5, qui correspondent aux valeurs renvoyées par X-Means, ce qui valide ce choix a posteriori.

Pour choisir le meilleur nombre de groupes, nous utilisons ensuite la méthode du coude, également appelée *scree test* en anglais. Elle consiste à repérer le point d'inflexion de la courbe d'erreur. Cela signifie qu'augmenter le nombre de groupe ne diminue plus autant l'erreur de partitionnement. Cette méthode est illustrée dans la figure 5.12.

Nous étudierons ensuite, pour ce nombre optimal de groupes, les détails des indicateurs pour les élèves du groupe, afin de tenter de mieux comprendre la composition et la signification de ces groupes.

5.4.3 Jeux de données

Nous avons choisi 3 formations différentes, dont deux promotions différentes d'une même formation, que nous appellerons formations A1 et A2, et une formation différente B. La formation A1 a 56 élèves, A2 en a 15 et B en a 30. A1 et A2 durent environ un an, tandis que B dure trois mois.

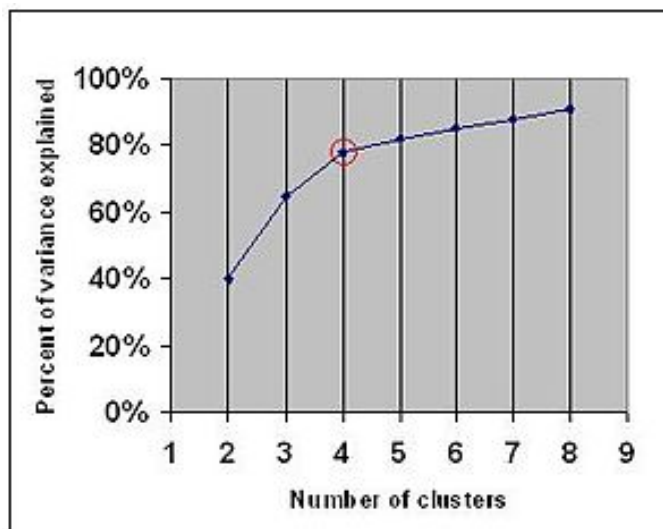


FIGURE 5.12 – La méthode du coude illustrée

5.4.4 Résultats

5.4.4.1 Nombre optimal de groupes

Les figures 5.13 et 5.14 montrent les résultats des quatre algorithmes sur nos trois jeux de données. Le premier montre le nombre moyen de groupes proposé par X-Means pour décrire au mieux nos données. Les trois autres graphes montrent l'erreur moyenne obtenue en fonction du nombre de groupes fixé pour les algorithmes K-Means, Hierarchical clustering et Expectation Maximisation.

Pour la formation A1 (courbe rouge en trait plein), X-Means propose 2 groupes, ce qui est aussi le choix fait par Expectation Maximisation. Cependant, Hierarchical Clustering se trompe et propose un groupe composé seulement du meilleur de tous les élèves, sans que nous comprenions vraiment pourquoi.

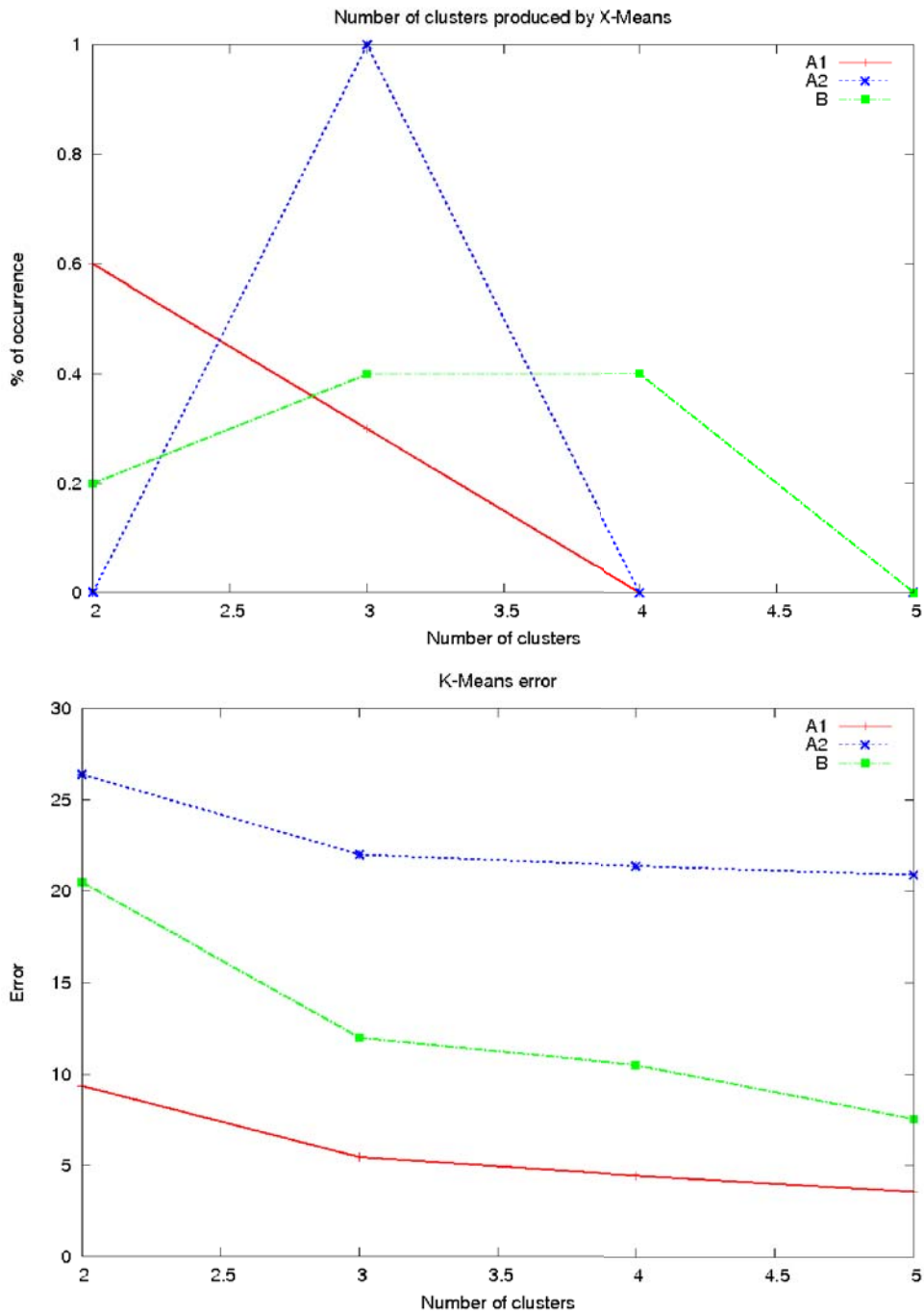


FIGURE 5.13 – Nombre optimal de groupes trouvé par chaque algorithme pour nos trois formations de test

La formation A2 (en bleu pointillé) produit des courbes très plates et peu significatives sauf avec K-Means, dont le net point d'inflexion à 3 groupes correspond au résultat proposé par X-Means (ce qui est normal dans la mesure où X-Means est

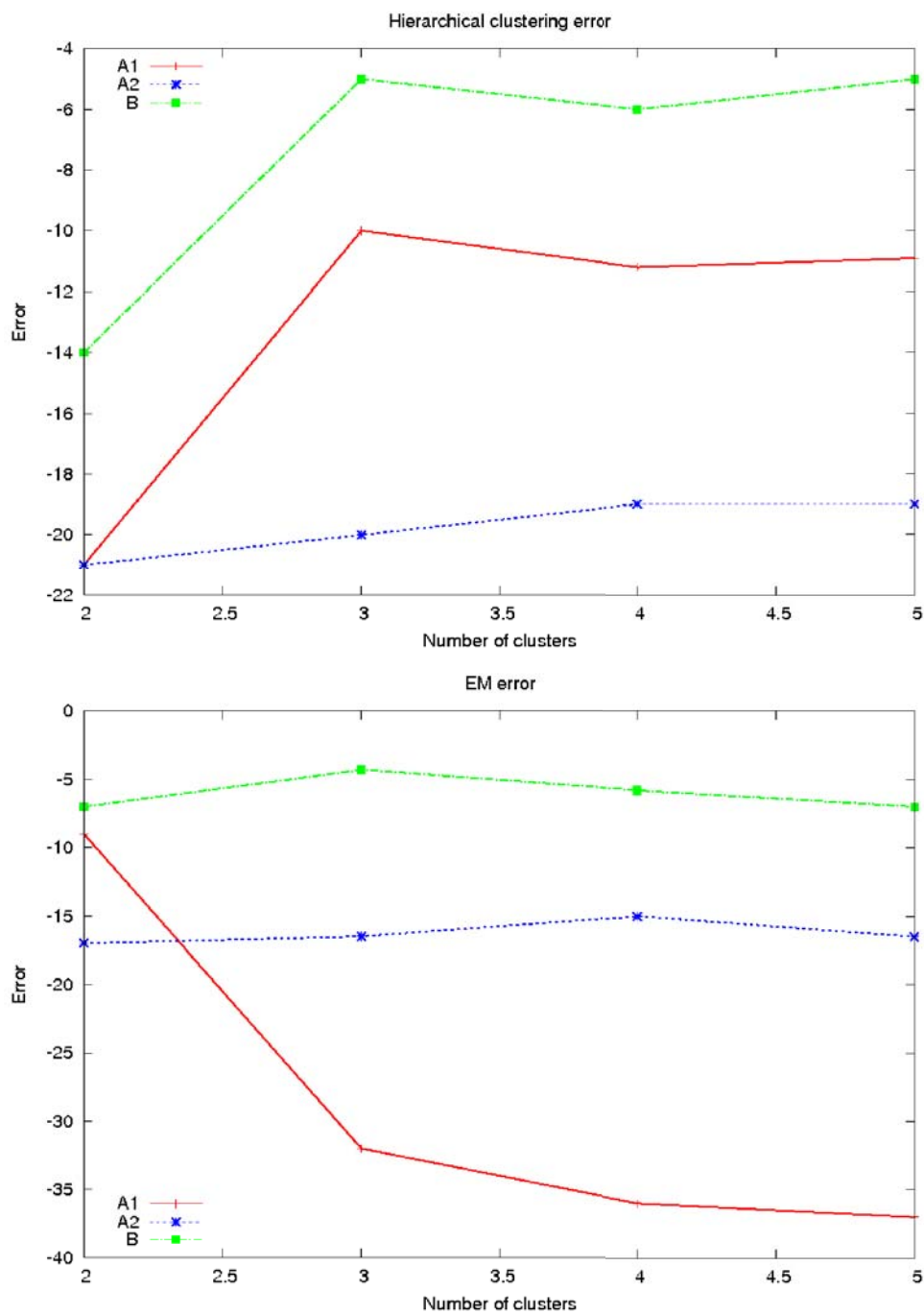


FIGURE 5.14 – Nombre optimal de groupes trouvé par chaque algorithme pour nos trois formations de test (suite)

basé sur K-Means, même s'il propose en plus un "bon" nombre de groupes).

Pour la formation B, en vert avec des tirets, les quatre méthodes s'accordent pour proposer 3 groupes.

5.4.4.2 Signification des groupes

À notre grande surprise, les groupes que nous avons observés pour les trois formations sélectionnées ne montraient guère de distinction plus pertinente que des variations quantitatives entre les élèves plus ou moins actifs, et ce quel que soit le nombre de groupes sélectionné. Par exemple, nous n'avons pas eu de groupe se distinguant d'un autre par son taux d'activité sur le forum comme [Taylor, 2002], ni de groupes d'élèves obtenant de bonnes notes malgré une faible activité. Il n'y a eu qu'une formation où certains paramètres permettaient de séparer d'un groupe de bons élèves un sous-groupe où ces élèves faisaient moins d'activités de type devoir. Un exemple de groupes obtenus est présenté dans la figure 5.15.

Quand le nombre de groupes requis augmentait à 4 ou 5, plutôt que de voir émerger ce genre de structure, nous obtenions souvent un groupe composé d'un seul élève, souvent le meilleur ou le moins bon de sa formation. Ceci indique que nous atteignons un nombre de groupes trop grand et ne permettant pas bien de représenter les données, et que l'augmenter encore n'aurait pas servi à dégager davantage d'information.

Pour expliquer ce phénomène, nous proposons les raisons suivantes :

- un nombre d'étudiants relativement faible par formation (entre 15 et 56), qui peut présenter moins de variété de comportement ;
- une formation ciblée vers un public peut-être assez homogène en termes d'âge, d'expérience professionnelle, et d'habitudes d'utilisation des TICE ;
- un effet de cercle vicieux pouvant avoir lieu sur le forum faisant que si personne ne se met à l'utiliser, personne d'autre n'aura d'incitation à s'en servir. Nous avons effectivement remarqué parmi nos élèves une tendance à utiliser surtout le forum pour communiquer avec les enseignants plus qu'entre eux.

Cependant, nous avons relevé que la distinction faite entre les élèves pour des nombres de groupes moins élevés était tout à fait pertinente par rapport à notre problématique visant à repérer des élèves en situation d'échec potentiel, surtout dans la mesure où nos résultats semblent indiquer une bonne corrélation entre l'activité globale et les notes obtenues aux activités notées. Utiliser des algorithmes de partitionnement de données est donc bien une solution pertinente à notre problème de suivi des étudiants.

Nous avons présenté ces expériences et leurs résultats dans les articles [Bovo et al., 2013a, Bovo et al., 2013b, Bovo et al., 2013c].

	Lecture de leçons	Lecture de PDFs	Consultation de ressources	Fréquence de connexion	Dernière connexion	Quiz effectués	SCORMs effectués	Devoirs effectués
1	2,84	5,549	5,044	4,182	6,968	0	3,062	2,802
2	8,281	6,668	8,172	9,234	9,402	0	8,193	7,358
3	6,437	7,574	5,284	6,975	8,927	0	5,725	5,57

	Hot Potatoes effectués	Notes obtenues	Dernières notes obtenues	Meilleures notes obtenues	Lecture du forum	Nouveaux sujets sur le forum	Réponses sur le forum	Temps passé en ligne
	3,664	0,37	0,37	0,37	3,139	1,389	0,741	4,192
	7,765	0	0	0	8,747	5,048	3,678	6,32
	6,526	0,435	0,435	0,435	4,311	1,833	1,413	6,989

FIGURE 5.15 – Un exemple de résultat de partitionnement ne présentant que des différences quantitatives

Dans ce chapitre, nous avons donc montré comment nos diverses propositions pouvaient s'agencer entre elles pour donner naissance à une application web répondant aux besoins en terme de suivi des élèves. Nous avons ensuite utilisé cette application pour mener des expériences de partitionnement sur des données issues de formations réelles. Les premiers résultats fournis par ce partitionnement de données sont surprenants, puisqu'ils semblent montrer une corrélation quasi-parfaite entre la quantité d'activité dans tous les domaines ainsi qu'avec les résultats aux activités notées. Toutefois, ils semblent également montrer la pertinence de notre proposition d'utilisation du partitionnement pour créer des groupes d'élèves au comportement similaire, et se baser ensuite sur ces groupes pour améliorer la qualité du suivi proposé. Ces résultats nous poussent également à approfondir les tests permis par ce nouvel outil.

Chapitre 6

Conclusion et perspectives

Dans ce chapitre final, nous revenons sur le travail accompli et proposons des pistes de poursuite du projet.

6.1 Synthèse des contributions

À ses débuts, notre projet avait été lancé avec plusieurs objectifs. Le premier était de recenser les apports existants de l'apprentissage automatique dans le domaine de l'e-learning et de chercher à en développer de nouveaux. Le deuxième était de créer un modèle et un moteur de prise de décision automatique pour des tuteurs virtuels basé sur la prédiction de motifs et inspiré du fonctionnement du cerveau humain, se basant sur le modèle de représentation et de prédiction de données Hierarchical Temporal Memory. Le troisième était de développer des applications industrielles utilisant les outils proposés pour développer les possibilités offertes par les formation en e-learning.

Après une étude un peu plus poussée du domaine de l'apprentissage en ligne et de ses enjeux, nous avons pu préciser notre problématique. Notre objectif principal a alors été d'améliorer les possibilités de suivi des élèves en cours de formation en ligne pour éviter leur décrochage ou leur échec. Nous jugions cet objectif extrêmement important puisqu'il avait la possibilité d'impacter directement la situation de bon nombre d'élèves et de responsables de formation. Notre deuxième objectif dans le courant de cette thèse était d'étudier les possibilités de l'algorithme HTM-CLA pour nous aider à réaliser notre premier objectif. Nous avons présenté dans l'introduction de ce document trois idées différentes d'utilisation et adaptation possibles de cet algorithme.

Dans le courant de cette thèse, nous avons commencé par chercher à mieux définir les données qui étaient disponibles afin de pouvoir connaître le comportement des étudiants et donc les suivre. La section 2.1 présente les résultats de cette étude, qui a établi une différence entre les traces d'utilisation mémorisées par le LMS et les autres données stockées en des endroits divers, faisant émerger un besoin de centralisation de ces données. À partir de ces données, nous avons ensuite proposé dans la section 2.2 des indicateurs de comportement qui nous semblent à la fois

pertinents à notre projet et suffisamment génériques pour pouvoir être utilisés par d'autres. Ils sont résumés dans la figure 2.3.

Nous avons poursuivi par une revue des possibilités et des usages actuels de l'apprentissage automatique et de l'exploration de données, qui a été exposée dans la section 3.1. Dans la section 3.2, nous avons exposé ce qui était, à notre sens, les principaux usages, simples et relativement abstraits, qu'on pouvait faire de ces méthodes pour le suivi des élèves, répondant ainsi au premier objectif initial de notre projet, revu dans le cadre de notre problématique.

Conformément à notre deuxième objectif, nous nous sommes beaucoup intéressés pour ces fins à l'algorithme HTM-CLA. Nous avons tenté d'expliquer dans la partie 4.2 en quoi il se distinguait des algorithmes proches (réseaux de neurones, apprentissage profond) par son paradigme particulier basé sur la mémorisation de séquences temporelles de motif et leur prédiction, et sa modélisation basée sur les micro-colonnes corticales. Nous avons également expliqué en quoi nous pensions que ses hypothèses fondatrices étaient bien adaptées à notre problème dans la partie 4.1.2, justifiant ainsi notre intérêt pour cet algorithme.

Dans la section 4.3, nous détaillons nos propositions pour adapter HTM-CLA à des fonctionnalités d'apprentissage automatique classique, ce qui correspond à notre première idée d'utilisation. Nous détaillons notre proposition jusqu'au niveau de l'implémentation. Toutefois, nous n'avons pas réussi à obtenir des résultats probants à partir de cette implémentation, ce qui ouvre des perspectives pour des travaux futurs.

Dans la section 4.3.5, nous détaillons nos propositions pour adapter HTM-CLA à la génération de comportement, ce qui correspond à notre deuxième idée d'utilisation. Là encore, nous avons été jusqu'à présenter une implémentation de ces idées, qui n'a toutefois pas atteint non plus le stade fonctionnel.

Suivant notre troisième objectif initial, nous nous sommes alors attelés avec Andil à la conception et à la réalisation d'une application regroupant les fonctionnalités que nous avons définies : capable de centraliser toutes les données issues d'une formation en ligne, avec import automatique des données, avec présentation riche et visuellement très claire des ces données et de statistiques qui en seraient tirées, capable de calculer et afficher les indicateurs que nous avons définis, et d'effectuer les traitements d'apprentissage automatique. Nous avons expliqué dans la section 5.1 nos choix architecturaux et présenté dans la section 5.3 le résultat, à savoir une application web du nom de GIGA, déjà commercialisée et appréciée par les responsables de formation.

Nous avons, grâce à cette application, pu mener de premières expériences de partitionnement de données sur des données issues de formations réelles, grâce à notre institut de formation partenaire, dont nous présentons à la section 5.4 les résultats accompagnés d'une analyse et d'une tentative d'explication. Bien qu'un peu surprenants, ces tests semblent indiquer qu'un simple partitionnement permet dans bien des cas de révéler les étudiants en difficulté ou en voie d'abandon. Ce résultat valide donc notre approche et promet d'être très utile pour identifier et suivre ces élèves.

Nous avons donc pu étudier la majorité de nos objectifs et proposer pour chacun d'entre eux des solutions qui nous semblent à la fois bien adaptées et, pour certaines d'entre elles, assez innovantes. L'ensemble de nos propositions et leur articulation est résumée dans la figure 1.2. Toutefois, notre choix de sujet était quelque peu ambitieux pour le cadre d'une thèse, c'est pourquoi il reste encore beaucoup à faire, notamment du côté de l'implémentation et de la validation de nos propositions.

Cette thèse a également débouché sur plusieurs publications scientifiques et présentations en conférence, récapitulées dans l'annexe F.

6.2 Critiques et discussion

Nous avons débuté ce projet avec un sujet très vaste, dont nous n'avons pas eu l'occasion d'aborder tous les aspects. Notamment, nous avons identifié une autre problématique relevant davantage du domaine de l'ingénierie de la formation, qui nous paraît également très intéressante et prometteuse. Il s'agissait d'essayer d'améliorer la qualité de la formation proposée aux élèves. Bien qu'ayant identifié cette autre problématique assez tôt dans le courant du projet, nous avons jugé préférable de débiter par le suivi des étudiants, qui nous a semblé être un besoin plus important et plus urgent, en tout cas plus réclamé par les responsables de formation. Nous détaillerons nos pistes pour cette problématique dans la partie suivante.

Nous avons bien conscience que les résultats de tests que nous avons à présenter sont peu nombreux et donc peu pertinents. Toutefois, de futurs tests seront facilités par la création de GIGA et son interfaçage avec Weka, et pourraient être encore facilités par l'ajout de fonctionnalités comme l'import automatique des notes dans GIGA pour la classification et la régression.

Lorsque nous avons débuté notre projet, seul un rapport technique existait pour décrire HTM-CLA. Quelques implémentations non officielles circulaient, mais nous avons jugé qu'il serait quasiment aussi long de les comprendre pour vérifier qu'elles étaient correctes que d'en refaire, et qu'en outre la phase de développement nous permettrait de mieux comprendre et maîtriser le code.

Nous ne savons toujours pas si ces autres implémentations auraient mieux fonctionné que la nôtre, étant donné la difficulté qu'aurait demandé une comparaison, que nous n'avons donc pas cherché à réaliser, mais nous avons clairement sous-estimé l'ampleur de la tâche : en effet, pour passer du fonctionnement d'une unité au fonctionnement global, le phénomène d'émergence rend la compréhension du lien de l'un à l'autre très difficile.

Toutefois, depuis quelques mois, Numenta a rendu disponible à tous le code source de son implémentation de HTM-CLA, qu'ils proposent d'installer sous forme de librairie Python. Cette librairie se nomme NuPIC, pour Numenta Platform for Intelligent Computing. Le moteur est en C++, mais toutes les couches autour sont en Python. Il comprend un composant nommé Online Prediction Framework (OPF), qui permet, comme son nom l'indique, de réaliser des prédictions, ainsi qu'un système de classification. Enfin, il est distribué sous licence GPL 3.

Il pourrait donc être très intéressant d'essayer d'adapter nos propositions à leur implémentation. Du fait qu'elle soit officielle, elle offre plusieurs garanties intéressantes que les autres n'avaient pas :

- elle va continuer à être maintenue et développée par Numenta
- elle offre un fonctionnement correct, garanti par l'existence d'applications commerciales utilisables
- elle va bénéficier des développements de la communauté qui s'est rassemblée autour du projet.

Toutefois, l'utiliser aurait également présenté des contraintes pour nous, puisque la licence GPL est contaminante. Cela signifie que si nous avions eu besoin de modifier NuPIC pour l'adapter à nos besoins, nous n'aurions pas facilement pu utiliser cette version modifiée dans les applications d'Andil, comme GIGA. C'est également le cas de Weka, mais cela ne nous a pas posé problème, puisque nous n'avons pas eu besoin de la modifier et que nous n'en avons utilisé que l'API. Par prudence, nous avons tout de même prévu de séparer le code associé sous forme de greffon logiciel. Mais il n'était de toute façon pas envisageable de recoder entièrement Weka ou même le petit échantillon d'algorithmes que nous avons utilisé dans le cadre limité de notre projet, contrairement à HTM.

Toujours est-il qu'en l'absence d'implémentation HTM-CLA fonctionnelle, nous n'avons pas réellement pu tester nos propositions, ni pour l'adapter aux méthodes d'apprentissage automatique classiques, ni pour l'utiliser pour le suivi des étudiants avec ses fonctionnalités natives de prédiction, ni pour en faire la base du moteur de comportement. Toutefois, nos propositions nous semblent quand même pertinentes, et pourraient être adaptées à la nouvelle implémentation de HTM-CLA dont nous venons de parler. Si l'on souhaite poursuivre les recherches avec l'implémentation actuelle, il faudra améliorer l'interface graphique, voire même en concevoir une mieux adaptée.

Nous n'avons pas encore collaboré avec des chercheurs issus d'autres équipes dans le cadre de ce projet. Maintenant que notre travail a contribué à dégrossir notre projet, il serait très intéressant de chercher à travailler avec d'autres chercheurs spécialisés dans HTM, d'une part, et d'autre part dans d'autres parties du tuteur virtuel, comme peut-être la génération de langage si on décide d'en faire un agent conversationnel, ou la simulation d'émotions.

6.3 Perspectives

Nous allons ici mentionner les pistes que nous voyons pour la poursuite de notre projet.

6.3.1 Modélisation et évolution de nos indicateurs

Nous avons mentionné dans la partie 2.2 l'existence d'un langage de modélisation de traces d'utilisation et d'indicateurs, UTL ([Choquet et al., 2007]), ainsi que de son extension permettant également de décrire le calcul des indicateurs, UTL-CL ([Pham Thi Ngoc et al., 2009]). Une suite logique de notre travail de définition d'indicateurs serait d'essayer de modéliser nos indicateurs dans ce langage, afin de voir si ce processus est aisé et ce qu'il peut apporter ensuite en termes de généralité.

Nous ne considérons pas notre liste d'indicateurs comme figée. Tout indicateur semblant pertinent à notre problème et en couvrant un nouvel aspect pourrait bien entendu y être rajouté par la suite.

6.3.2 Perspectives d'évolution de GIGA

Une des suites les plus immédiates à donner à ce projet serait la réalisation de tests plus étendus utilisant la bibliothèque Weka, adaptée par nos soins à GIGA, et les données fournies par Juriscampus.

Nous pourrions répéter les expériences de partitionnement de données, mais en les étendant à toutes les formations dont nous disposons, pour voir si les résultats se maintiennent sur ce jeu de données plus large ou si certaines formations présentent de grandes différences au niveau de l'implication des élèves. Nous pourrions également les étendre à tous les algorithmes proposés par Weka, dont nous n'avons sélectionné que quelques-uns parmi les plus connus et représentatifs des différentes méthodes existantes.

Une fois que nous aurons automatisé l'import des notes aux examens finaux à GIGA, nous aurons plus de facilité à faire d'autres types de test : elles sont en effet nécessaires à nos propositions d'utilisation de méthodes de classification et de régression, servant respectivement, pour rappel, à chercher à prédire le succès ou l'échec des élèves à la formation, comprendre quels en sont les facteurs et prédire leur moyenne finale.

Cependant, tous ces calculs sont basés sur nos indicateurs, qui écrasent la dimension temporelle des données. Afin de la récupérer, nous proposons des tests basés sur des indicateurs qui ne recouvrent pas toute la période de temps de la formation, mais seulement un intervalle défini manuellement. En jouant sur cet intervalle, on peut obtenir plusieurs résultats intéressants. Si on le fait varier depuis le début de la formation jusqu'à un point donné dans l'année, on peut vérifier à partir de quel moment de l'année les prédictions acquièrent une justesse dépassant un certain seuil. Si on le fait varier semaine par semaine, on peut chercher à voir si certaines périodes de l'année sont plus représentatives que d'autres du résultat final.

On peut à cette fin également, bien évidemment, implémenter nos propositions de prédiction, ce que nous n'avons pas encore cherché à faire d'une autre façon qu'avec HTM. En effet, Weka ne propose pas d'algorithme de prédiction. Pour cela, on pourra se baser sur les algorithmes de type série temporelle décrits dans la partie 3.1.3.1.

Nous n'avons pas encore non plus proposé d'implémentation pour la détection d'anomalie, pensant là encore nous baser sur HTM, avec la méthode décrite dans la section 4.3. Là encore, il serait possible, si l'on voulait obtenir des résultats rapides, de rajouter à GIGA des algorithmes différents (Weka n'en propose pas non plus).

Toujours dans la même idée de substitution temporaire, on pourrait commencer à créer un tuteur virtuel intelligent sur la base d'un système de génération de comportement déjà prouvé, comme par exemple les systèmes de classeurs. Toutefois, il s'agit là d'un système d'apprentissage par renforcement, comme bon nombre d'autres systèmes de génération de comportement. Or, nous avons déjà mentionné qu'il ne semblait pas évident de récompenser une action de type guidage d'un élève ou remontée d'information vers un tuteur. Il faudrait donc définir, avant de pouvoir les employer, des exemples de test sur lesquels l'apprentissage devrait être restreint et quasiment supervisé, et espérer pouvoir les faire suffisamment variés pour qu'une généricité suffisante s'installe, ou poster un humain au rôle d'estimation de la récompense.

Nous avons également des perspectives plus simples, comme l'ajout des sources de données externes au LMS à GIGA, si possible de façon automatique, et l'import automatique des traces venant d'autres LMS que Moodle. En parallèle avec l'ajout de nouveaux types de données, nous devons implémenter le calcul des indicateurs correspondant et créer les nouvelles pages pour les afficher.

Nous prévoyons aussi de rajouter à GIGA de nouvelles fonctionnalités, notamment celle de la sélection facilitée de groupes d'élèves. Actuellement, le filtre permet de sélectionner facilement des groupes correspondant à une formation, ou à une option particulière au sein d'une formation. Mais il n'est pas encore possible de sélectionner un groupe d'élèves entièrement personnalisé, ni de conserver les groupes définis par l'algorithme de partitionnement, et pas non plus de sauvegarder ces groupes d'une fois sur l'autre. Cette évolution pourrait pourtant faciliter le suivi des élèves.

À plus long terme, nous envisageons également une plus grande intégration de GIGA avec les autres applications développées par Andil. L'une d'entre elles s'appelle le WOK (Wiki of Knowledge) et est un gestionnaire de contenus de cours comprenant un éditeur (WYSIWYG ou avec syntaxe wiki), une gestion de versions, des métadonnées, et un export permettant la publication ou la mise à jour manuelle ou planifiée sur les plate-formes LMS dans différents formats. Ainsi, pour les formations qui l'utilisent également, les données pédagogiques sont déjà informatisées et standardisées dans sa base de données.

Avoir accès aux données du WOK correspondant aux cours sur le LMS à analyser offrirait donc des données supplémentaires, notamment la date d'ouverture d'un cours ou une information concernant la modification d'un contenu pédagogique, que l'on ne retrouverait pas dans le LMS, mais permettant d'expliquer davantage les comportements des étudiants.

Pour l'instant, GIGA n'a pas été conçu comme étant directement ouvert aux élèves. En effet, cela poserait des problèmes au niveau de la confidentialité des données que de lui permettre d'avoir accès au niveau des autres étudiants. Or, si nos statistiques sont absolues, nos indicateurs sont relatifs les uns par rapport aux autres. Il ne s'agirait pas non plus de démoraliser un étudiant en le montrant comme systématiquement en queue de peloton. Néanmoins, la réflexion sur ce sujet peut rester ouverte, dans la mesure où la dimension sociale et compétitive de l'apprentissage et la comparaison avec les autres sont des facteurs prouvés de motivation [Festinger, 1954].

Toutefois, il semblerait quand même intéressant de permettre à l'étudiant de connaître son progrès et de pouvoir estimer s'il est suffisant, dans la mesure où cela a été prouvé être un facteur d'efficacité dans l'apprentissage [Schunk, 2003]. L'apprenant pourrait alors mesurer et comparer son activité selon des critères absolus définis par les responsables de formation ou tout simplement en termes de proportion d'activités réalisées. C'est ce qu'a fait [Vermeulen et al., 2014], qui propose un greffon pour le LMS intitulé Progress Tracker, et qui montre son efficacité sur l'engagement des élèves. Dans cette perspective, Andil a toutefois déjà publié un greffon pour Moodle permettant d'attribuer des médailles aux élèves.

6.3.3 Concrétiser nos propositions utilisant HTM

En l'absence d'implémentation fonctionnelle de HTM-CLA, nous n'avons pas réellement pu tester nos propositions. Bien évidemment, la correction des derniers problèmes de notre code et les tests associés à toutes nos propositions pourraient être les éléments les plus intéressants d'une suite à apporter à ce projet, même s'il ne s'agirait sûrement pas d'une tâche facile.

Pour pouvoir tester nos propositions en garantissant un moteur HTM correct, elles pourraient être adaptées à NuPIC, la nouvelle implémentation de HTM-CLA dont nous venons de parler.

Si l'on souhaite poursuivre les recherches avec l'implémentation actuelle, il faudra absolument améliorer l'interface graphique de visualisation du fonctionnement de la hiérarchie, voire même en concevoir une mieux adaptée à la tâche extrêmement complexe qu'est le débogage du code et le réglage de ses paramètres.

En parlant du réglage des paramètres, nous avons appris lors de l'ouverture de NuPIC au public qu'ils utilisaient à cet effet un algorithme d'optimisation par essais particuliers (en anglais *Particle swarm optimization*). Nous avons déjà pensé pour cela à l'utilisation d'un algorithme génétique. Toutefois, nous avons pensé que le temps de calcul associé à ce type d'optimisation serait rédhibitoire. NuPIC étant plus rapide que notre implémentation, cela ouvre des possibilités de ce côté.

Numenta est en train actuellement de développer un partenariat avec cortical.io, une société autrichienne qui souhaite appliquer HTM au traitement du langage naturel. Ainsi, il est possible qu'il devienne bientôt aisé d'apprendre et même de comprendre un texte donné comme entrée à un HTM pré-entraîné sur le domaine de vocabulaire utilisé. Si on construisait alors une nouvelle hiérarchie HTM visant à apprendre les contenus pédagogiques textuels, on pourrait lui faire construire une

représentation de toutes les connaissances contenues dans les cours, avec, pourquoi pas, des applications de type moteur de recherche ou analyse de concept. Néanmoins, ce contenu n'est pas pour l'instant disponible dans GIGA et n'a pas vraiment de raison d'y figurer. Il s'agit donc a priori d'un projet indépendant.

6.3.4 Le domaine de l'ingénierie de formation

Lors de notre étude de début de projet, nous avons découvert une autre problématique, celle de l'ingénierie de formation, qui nous paraît également très intéressante et qui pourrait soit être le thème d'un deuxième volet futur de notre projet.

Un premier objectif lié à cette problématique serait d'améliorer les formations mises à disposition des élèves. Il serait très intéressant de repérer qu'un quiz est plus difficile que prévu ou qu'une leçon ne retient pas l'intérêt des étudiants qui ne la lisent pas jusqu'au bout afin de pouvoir les améliorer en conséquence, ou si les étudiants ont besoin de poser beaucoup de questions sur le forum, signe que des explications manquent peut-être dans les contenus ou les consignes. Or, pour cela, les données à examiner sont également les traces d'utilisation fournies par le LMS, si l'on ne veut pas solliciter les élèves. Si l'on souhaite leur demander leurs retours, on peut également prévoir de centraliser ce nouveau type de données dans GIGA.

Dans cette perspective, l'intégration avec le WOK est encore plus utile puisqu'elle permet d'accéder aux maquettes pédagogiques, à des statistiques sur les contenus pédagogiques, à leur changement de version et à d'autres données de ce type.

Nous souhaitons également évaluer de manière plus précise l'importance des connaissances jugées comme pré-requises à la formation. Le niveau des élèves dans ces disciplines a-t-il réellement un impact sur leur succès ? Si l'on réussit à l'évaluer au préalable, on pourra chercher à quantifier cet effet. Cela peut permettre, si l'on se rend compte qu'un pré-requis n'est en fait pas important, d'ouvrir la formation à un public plus large, ou au contraire, s'il a trop d'impact, d'éviter à des élèves ne remplissant pas ce critère de s'engager inutilement dans la formation.

À ce sujet, une autre application récemment créée par Andil est un profil de compétences, qui permet d'évaluer un élève dans différentes disciplines jugées nécessaires à sa formation. Un lien avec cette application permettrait soit d'avoir une évaluation préliminaire des élèves avant leur formation, permettant d'étudier l'influence de leur niveau préalable dans leur succès ; soit une évaluation automatisée a posteriori, déduisant leur nouveau niveau dans les compétences associées à la formation.

Enfin, nous souhaiterions aussi évaluer l'effet de l'ordre de parcours des ressources pédagogiques pendant la formation sur le succès des élèves. Idéalement, cela nous servirait à extraire les parcours suivis par les meilleurs élèves de chaque formation et voir s'ils diffèrent de ceux suivis par les moins performants pour juger de l'importance de cet ordre. On pourrait alors faire une recommandation aux élèves concernant le parcours optimal.

Une fois des indicateurs et statistiques, et des méthodes de fouille de données, proposés pour ces trois thèmes, on pourrait assez facilement en programmer l'implémentation dans GIGA, sans changer les données qui sont récoltées. On pourrait

alors avoir une troisième partie du site dédiée à l'analyse des contenus, voire une quatrième orientée par auteur de contenu, et une autre partie dédiée aux programmes, qui présenterait les parcours et les pré-requis. Toutefois, on peut également envisager d'en faire une application différente utilisant la même base de données, ou une duplication de la base de données de GIGA.

Nous pourrions également implémenter dans GIGA les indicateurs proposés dans [Dringus and Ellis, 2005] pour évaluer l'activité d'un forum, qui permettent aux responsables de formation d'évaluer s'il est assez actif et de prendre des mesures pour le rendre plus actif le cas échéant. Nous avons vu que c'était un problème assez fréquent dans nos formations.

Enfin, on pourrait, si l'on détectait des groupes d'élèves de comportement différent, leur proposer un parcours ou des ressources mieux adaptées à leur style d'apprentissage, afin de personnaliser leur cursus. Mais cela représente beaucoup plus de travail, et doit s'implémenter directement au niveau du LMS.

6.3.5 Éthique de la surveillance

De façon quasi systématique, à chaque fois que nous avons parlé de nos travaux à un public hors du domaine, on nous a demandé : « Mais... c'est du Big Brother, alors ? » Le grand public semble avoir été fortement sensibilisé au problème de l'utilisation abusive des données à des fins pas forcément bienveillantes. Nous les avons à chaque fois rassurés en disant que nous travaillions pour le bien de l'élève surveillé, et qu'avant nos travaux, ces données étaient tout de même récoltées, et certaines d'entre elles analysées manuellement. En effet, personne n'aurait rien à gagner à ce qu'un élève ne quitte une formation avant sa fin : il ne peut donc pas s'agir de chercher à éliminer au plus vite les derniers de la classe.

Toutefois, il pourrait sembler légitime de mener une réflexion plus profonde sur l'éthique de cette utilisation des données personnelles des étudiants. On pourrait imaginer une solution temporaire en attendant les résultats de cette réflexion qui consisterait à veiller à ce que les étudiants soient informés de cette collecte et de cette analyse de données, et peut-être même à leur donner l'option de ne pas souscrire à l'analyse automatique, s'ils précisent qu'ils comprennent bien qu'ils recevront en retour un suivi de moins bonne qualité.

6.4 Conclusion

Durant ce doctorat, nous avons eu le privilège d'une double perspective. D'un côté, les enjeux très concrets d'un contexte industriel très demandeur d'applications innovantes ; de l'autre côté, la stimulation intellectuelle de l'étude encore très fondamentale d'un algorithme novateur et qui nous semble toujours très prometteur, malgré le manque de résultats concrets que nous pouvons présenter à son sujet. Si ces deux points de vue nous ont parfois semblé comme menant nécessairement à un grand écart, nous pensons que la vision de chaque aspect a été enrichie par les contraintes de l'autre.

Le domaine de l'apprentissage en ligne, en pleine expansion, est amené à une vraie démocratisation qui verra son intégration à l'école, en formation supérieure, en formation continue, en entreprise... Mais son vrai défi est d'arriver à conserver un aspect humain. En dehors des formations mixtes, le suivi est nécessaire pour conserver ce lien crucial pour la motivation de l'apprenant. Mais l'échelle de ces formations rend nécessaire le travail d'automatisation partielle que nous entamons à peine. Bien loin de déshumaniser le suivi, il libère du temps à ceux qui l'effectuent pour apporter leur expertise et entrer en contact direct avec ceux qui en ont le plus besoin.

Avec HTM, nous pensons voir s'ouvrir devant nous un nouveau chapitre de l'histoire de l'intelligence artificielle. Il nous paraît enfin possible de toucher du doigt le fonctionnement du cerveau humain, ramené à sa fonctionnalité la plus basique, et non simulé dans un excès de détails comme le font actuellement les projets Blue Brain et SyNAPSE. Même si l'on voit bien les limites de ce modèle, encore incapable de gérer la motivation ou l'émotion, il nous semble ouvrir les portes des domaines désignés par le paradoxe de Moravec comme les plus difficiles de l'intelligence artificielle, et pourquoi pas celle de l'intelligence artificielle générale, lorsqu'une infrastructure matérielle mieux adaptée, probablement à base de memristors, lui permettra une mise à l'échelle plus efficace de ses capacités.

Notre domaine de travail est doublement informatique, aussi bien dans les outils et méthodes proposées que dans notre domaine d'application. Pourtant, c'est bien l'humain qui est au cœur de cette thèse, puisqu'il en inspire tout à la fois les besoins, les méthodes et les enjeux.

Annexe A

Détails de l'application GIGA

Nous présentons ici quelques captures d'écran de notre application GIGA.



FIGURE A.1 – La page d'accueil de GIGA, centrée sur une formation dont on voit l'IGA

En outre, nous rajoutons ici quelques témoignages de clients de GIGA. Ils sont authentiques, mais leurs auteurs ont demandé l'anonymat auprès d'Andil.

« Devant le grand nombre de stagiaires à suivre, j'ai été submergé par la quantité de travail. Je cherchais un outil qui pouvait me faire gagner du temps et j'ai trouvé GIGA. Il me facilite le suivi que je dois faire régulièrement sur les formations que nous avons en ligne. »

« Je cherchais un outil comme GIGA pour satisfaire le service des formations en ligne dans notre université. Nous avons pris la décision d'acheter la licence et nous avons personnalisé l'outil pour qu'il corresponde totalement à nos besoins. »

Etudiants	Lecture de leçons	Lecture de PDFs	Consultation de ressources	Fréquence de connexion	Dernière connexion	Quiz effectués	SCORMs effectués	Devoirs effectués	Hot Potatoes effectués	Notes obtenues	Dernières notes obtenues	Meilleures notes obtenues	Lecture du forum	Nouveaux sujets sur le forum	Réponses sur le forum
Arlandon Jean-Louis	2,43	10	4,06	2,85	9,63	0	2,37	0	4,86	5,35	4,62	5,61	2,55	0	0
BESSET Christophe	0,13	0,12	0,41	0,19	3,08	0	0,2	0	1,53	0	0	0	0,2	0	0
CANSSE Gunther	1,12	7,05	0,49	1,99	9,68	0	0	0	1,53	0	0	0	1,7	2,61	0
CASZALOT Philippe	5,79	7,78	3,76	4,95	9,68	0	8,41	4,07	8,89	7,53	9,21	9,16	4,35	0,43	0
CORNIERE Julie	0,26	1,84	1,77	0,81	2,83	0	1,74	1,37	7,5	0	0	0	0,13	0	0
CRONERT Michel	3,13	8,17	10	7,12	10	0	5,02	0,44	7,92	6,13	0,86	6,43	1,92	1,74	0
DEVAL Marvin	0,13	0,12	0,35	0,08	1,99	0	0	0	0	0	0	0	0,14	0	0
DEMAZOT Philippe	7,33	8,03	2,04	3,09	7,73	0	4,5	1,45	4,72	4,57	2,66	6,21	1,46	1,3	0
HERMANN Raphaël	0,72	7,87	2,66	1,51	9,32	0	8,67	0	8,89	8,08	9,33	9,52	2,06	0	0
HUGUET Eric	10	9,59	5,06	10	10	0	9,43	7,06	10	7,09	8,82	8,83	10	10	0
LORENZONE Jessica	3,57	8,22	5,49	3,92	8,88	0	3,59	0	6,25	3,52	2,05	5,05	3,53	3,91	0

FIGURE A.2 – Les différents indicateurs et leur valeur pour tous les étudiants de la formation sélectionnée

Etudiants	Présence	Étude	Activités	Forums	Notes
Arlandon Jean-Louis	4,71	5,5	2,07	0,85	5,19
BESSET Christophe	1,13	0,22	0,49	0,07	0
CANSSE Gunther	4,27	2,88	0,44	1,44	0
CASZALOT Philippe	5,87	5,78	6,11	1,59	8,63
CORNIERE Julie	1,33	1,29	3,03	0,04	0
CRONERT Michel	6,44	7,1	3,82	1,22	4,47
DEMAZOT Philippe	0,71	0,2	0	0,05	0
DEVAL Marvin	4,2	5,8	3,05	0,92	4,48
HENNI Abdelgani	3,98	3,75	5,02	0,69	8,98
HERMANN Raphaël	10	8,22	7,57	6,67	8,25
HUGUET Eric	5,03	5,76	2,81	2,48	3,54
JurisCampus apprenant	1,81	0,15	0,57	0,03	2,49
LORENZONE Jessica	5,87	7,73	8,41	3,24	10
Papret Laurent	1,83	2,03	2,67	0,2	0
SAINT LEGER Carine	4,86	6,83	5,73	3,68	9,39

FIGURE A.3 – Les différents indicateurs thématiques

The screenshot displays the 'Présence' (Presence) section of the GIG7 application. The table lists student names, their connection counts, weekly averages, last connection dates, and total online time.

Etudiant	Nombre de connexions	Moyenne / semaine	Date de dernière connexion	Temps passé en ligne
Arlandon Jean-Louis	106	2.0	jeudi 28 février 2013 18:52:02	194 heures 40 minutes et 51 secondes
BESSET Christophe	7	0.13	mardi 31 juillet 2012 17:19:00	12 heures 38 minutes et 17 secondes
CANSSE Gunther	74	1.4	vendredi 01 mars 2013 12:53:43	131 heures 24 minutes et 29 secondes
CASZALOT Philippe	184	3.47	vendredi 08 mars 2013 07:47:01	347 heures 51 minutes et 23 secondes
CORNIERE Julie	30	0.57	dimanche 15 juillet 2012 19:20:13	40 heures 54 minutes et 46 secondes
CRONERT Michel	265	5.0	jeudi 07 mars 2013 12:04:22	258 heures 14 minutes et 23 secondes
DEMAY Benjamin	3	0.06	vendredi 11 mai 2012 03:36:19	6 heures 23 minutes et 1 seconde
DEVAL Marvin	115	2.17	mardi 12 mars 2013 23:29:35	208 heures 3 minutes et 58 secondes
HENNI Abdelgani	56	1.06	vendredi 22 février 2013 19:07:40	130 heures 59 minutes et 9 secondes

FIGURE A.4 – Les différentes statistiques relevant du thème de l'étude de la présence en ligne

The screenshot displays the 'Etude' (Study) section of the GIG7 application. The table provides detailed statistics on lessons read, PDFs downloaded, resources opened, and study time for each student.

Etudiant	Nombre de leçons lues		Nombre de PDF téléchargés		Nombre de ressources ouvertes		Temps passé à étudier
	Distincts	Total	Distincts	Total	Distincts	Total	
Arlandon Jean-Louis	15	53	64	267	130	366	194 heures 40 minutes et 51 secondes
BESSET Christophe	1	1	1	1	15	18	12 heures 38 minutes et 17 secondes
CANSSE Gunther	8	13	56	79	16	42	131 heures 24 minutes et 29 secondes
CASZALOT Philippe	39	93	62	86	133	211	347 heures 51 minutes et 23 secondes
CORNIERE Julie	2	2	15	17	65	75	40 heures 54 minutes et 46 secondes
CRONERT Michel	21	51	63	111	358	520	258 heures 14 minutes et 23 secondes

FIGURE A.5 – Les différentes statistiques relevant du thème de l'étude des contenus

Filter: Juriscampus CGPS 12 01/03/2012 08/03/2013

Étudiant	Moyenne des dernières notes	Moyenne des meilleures notes	Moyenne de toutes les notes
Arlandon Jean-Louis	9 / 20	11 / 20	9.5 / 20
BESSET Christophe	-	-	-
CANSSE Gunther	-	-	-
CASZALOT Philippe	17.96 / 20	17.96 / 20	13.38 / 20
CORNIERE Julie	-	-	-
CRONERT Michel	1.67 / 20	12.61 / 20	10.89 / 20
DEMAY Benjamin	-	-	-
DEVAL Marvin	5.18 / 20	12.18 / 20	8.12 / 20
HENNI Abdelgani	18.19 / 20	18.67 / 20	14.35 / 20
HERMANN Raphaël	17.19 / 20	17.3 / 20	12.6 / 20
HUGUET Eric	4 / 20	9.9 / 20	6.25 / 20
JurisCampus apprenant	4.7 / 20	4.7 / 20	4.7 / 20
LORENZONE Jessica	19.49 / 20	19.6 / 20	17.77 / 20
Papret Laurent	-	-	-
SAINTE LÉGER Carine	18.97 / 20	19.07 / 20	15.46 / 20

FIGURE A.6 – Les moyennes des notes obtenues par l'élève

Étudiant	Sujets lancés		Réponses écrites		Discussions lues		Temps passé sur les forums
	Total	Distincts	Total	Distincts	Total	Distincts	
Arlandon Jean-Louis	0	0	0	0	36	60	194 heures 40 minutes et 51 secondes
BESSET Christophe	0	0	0	0	3	3	12 heures 38 minutes et 17 secondes
CANSSE Gunther	6	0	0	0	25	31	131 heures 24 minutes et 29 secondes
CASZALOT Philippe	1	0	0	0	60	117	347 heures 51 minutes et 23 secondes
CORNIERE Julie	0	0	0	0	2	2	40 heures 54 minutes et 46 secondes
CRONERT Michel	4	0	0	0	28	36	258 heures 14 minutes et 23 secondes

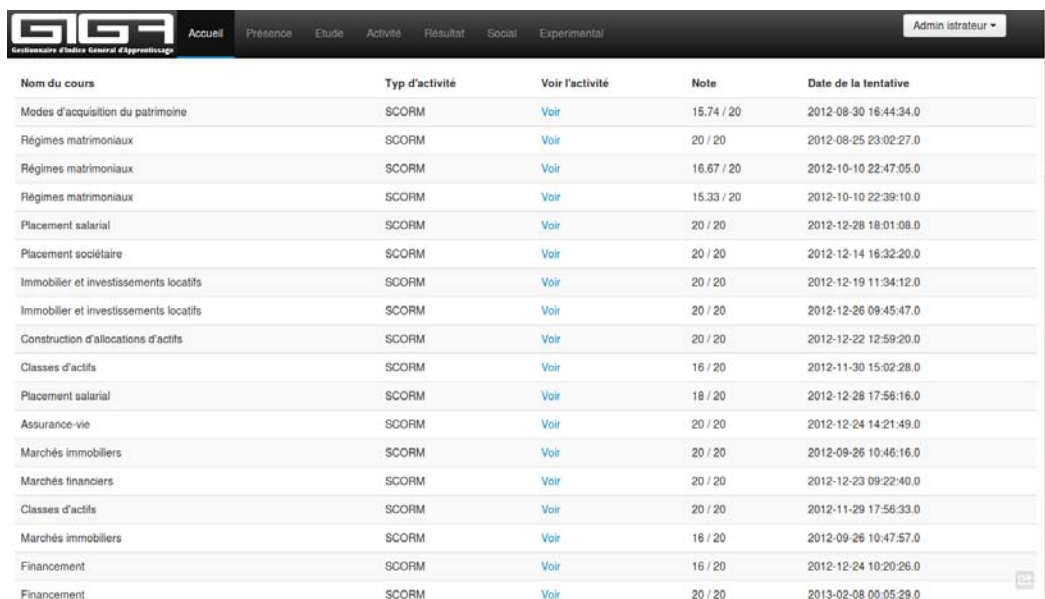
FIGURE A.7 – Les différentes statistiques relevant du thème de l'étude de l'interaction sociale

Etudiant	Quiz (sur 0 disponibles)		SCORM (sur 482 disponibles)		Devoir en ligne (sur 316 disponibles)		Hot Potatoes (sur 45 disponibles)		Temps passé sur les activités
	Distincts	Total	Distincts	Total	Distincts	Total	Distincts	Total	
Arlandon Jean-Louis	0	0	11	18	0	0	3	5	194 heures 40 minutes et 51 secondes
BESSET Christophe	0	0	1	1	0	0	1	1	12 heures 38 minutes et 17 secondes
CANSSE Gunther	0	0	0	0	0	0	1	1	131 heures 24 minutes et 29 secondes
CASZALOT Philippe	0	0	30	154	9	11	5	14	347 heures 51 minutes et 23 secondes
CORNIERE Julie	0	0	7	24	3	4	4	14	40 heures 54 minutes et 46 secondes
CRONERT Michel	0	0	22	51	1	1	5	7	258 heures 14 minutes et 23 secondes

FIGURE A.8 – La participation aux activités notées

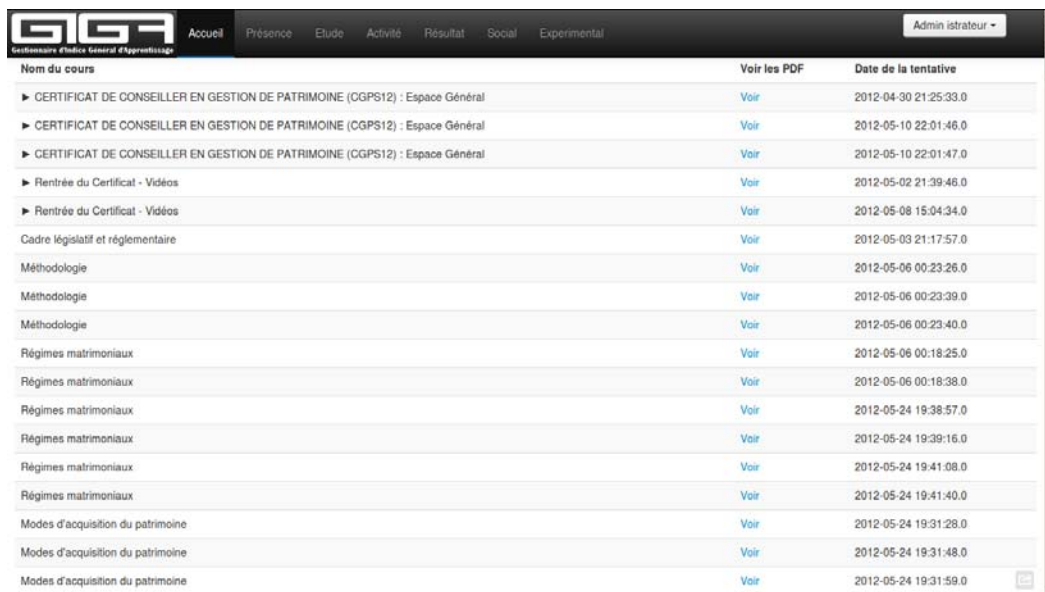
Nom du cours	Voir les Discussion dans les forum	Date de la tentative
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-05-09 21:51:03.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-05-09 21:51:46.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-05-10 22:05:19.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-08-11 11:07:08.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-09-10 20:34:07.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-09-10 20:38:29.0
► PROJET TUTORÉ	Voir	2012-09-04 21:21:16.0
► PROJET TUTORÉ	Voir	2012-09-04 21:21:48.0
► PROJET TUTORÉ	Voir	2012-09-10 20:45:29.0
► PROJET TUTORÉ	Voir	2012-09-10 20:45:43.0
► PROJET TUTORÉ	Voir	2012-09-23 10:03:20.0
► PROJET TUTORÉ	Voir	2012-09-23 10:04:06.0
► PROJET TUTORÉ	Voir	2012-09-23 10:04:43.0
► PROJET TUTORÉ	Voir	2012-09-24 08:32:59.0
► PROJET TUTORÉ	Voir	2012-09-24 10:57:28.0
► PROJET TUTORÉ	Voir	2012-09-25 11:57:50.0
Transmission du patrimoine	Voir	2012-08-11 12:16:48.0
Fiscalité des particuliers	Voir	2012-09-10 22:04:39.0

FIGURE A.9 – Le détail de l'activité d'un élève en lecture de forum



Nom du cours	Typ d'activité	Voir l'activité	Note	Date de la tentative
Modes d'acquisition du patrimoine	SCORM	Voir	15.74 / 20	2012-08-30 16:44:34.0
Régimes matrimoniaux	SCORM	Voir	20 / 20	2012-08-25 23:02:27.0
Régimes matrimoniaux	SCORM	Voir	16.67 / 20	2012-10-10 22:47:05.0
Régimes matrimoniaux	SCORM	Voir	15.33 / 20	2012-10-10 22:39:10.0
Placement salarial	SCORM	Voir	20 / 20	2012-12-28 18:01:08.0
Placement sociétaire	SCORM	Voir	20 / 20	2012-12-14 16:32:20.0
Immobilier et investissements locatifs	SCORM	Voir	20 / 20	2012-12-19 11:34:12.0
Immobilier et investissements locatifs	SCORM	Voir	20 / 20	2012-12-26 09:45:47.0
Construction d'allocations d'actifs	SCORM	Voir	20 / 20	2012-12-22 12:59:20.0
Classes d'actifs	SCORM	Voir	16 / 20	2012-11-30 15:02:28.0
Placement salarial	SCORM	Voir	18 / 20	2012-12-28 17:56:16.0
Assurance-vie	SCORM	Voir	20 / 20	2012-12-24 14:21:49.0
Marchés immobiliers	SCORM	Voir	20 / 20	2012-09-26 10:46:16.0
Marchés financiers	SCORM	Voir	20 / 20	2012-12-23 09:22:40.0
Classes d'actifs	SCORM	Voir	20 / 20	2012-11-29 17:56:33.0
Marchés immobiliers	SCORM	Voir	16 / 20	2012-09-26 10:47:57.0
Financement	SCORM	Voir	16 / 20	2012-12-24 10:20:26.0
Financement	SCORM	Voir	20 / 20	2013-02-08 00:05:29.0

FIGURE A.10 – Le détail de l'activité d'un élève en notes obtenues



Nom du cours	Voir les PDF	Date de la tentative
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-04-30 21:25:33.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-05-10 22:01:46.0
► CERTIFICAT DE CONSEILLER EN GESTION DE PATRIMOINE (CGPS12) : Espace Général	Voir	2012-05-10 22:01:47.0
► Rentrée du Certificat - Vidéos	Voir	2012-05-02 21:39:46.0
► Rentrée du Certificat - Vidéos	Voir	2012-05-08 15:04:34.0
Cadre législatif et réglementaire	Voir	2012-05-03 21:17:57.0
Méthodologie	Voir	2012-05-06 00:23:26.0
Méthodologie	Voir	2012-05-06 00:23:39.0
Méthodologie	Voir	2012-05-06 00:23:40.0
Régimes matrimoniaux	Voir	2012-05-06 00:18:25.0
Régimes matrimoniaux	Voir	2012-05-06 00:18:38.0
Régimes matrimoniaux	Voir	2012-05-24 19:38:57.0
Régimes matrimoniaux	Voir	2012-05-24 19:39:16.0
Régimes matrimoniaux	Voir	2012-05-24 19:41:08.0
Régimes matrimoniaux	Voir	2012-05-24 19:41:40.0
Modes d'acquisition du patrimoine	Voir	2012-05-24 19:31:28.0
Modes d'acquisition du patrimoine	Voir	2012-05-24 19:31:48.0
Modes d'acquisition du patrimoine	Voir	2012-05-24 19:31:59.0

FIGURE A.11 – Le détail de l'activité d'un élève en leçons lues sous forme de PDF

Nom du cours	Voir les SCORM	Date de la tentative
Régimes matrimoniaux	Voir	2012-05-06 13:28:26.0
Régimes matrimoniaux	Voir	2012-05-06 13:31:27.0
Régimes matrimoniaux	Voir	2012-05-06 13:32:04.0
Régimes matrimoniaux	Voir	2012-05-07 14:52:51.0
Régimes matrimoniaux	Voir	2012-05-10 16:45:30.0
Régimes matrimoniaux	Voir	2012-05-15 12:05:15.0
Régimes matrimoniaux	Voir	2012-05-18 10:04:56.0
Régimes matrimoniaux	Voir	2012-05-21 11:21:48.0
Régimes matrimoniaux	Voir	2012-05-29 16:15:15.0
Régimes matrimoniaux	Voir	2012-07-10 16:44:01.0
Régimes matrimoniaux	Voir	2012-07-10 16:46:03.0
Régimes matrimoniaux	Voir	2012-07-10 16:49:03.0
Régimes matrimoniaux	Voir	2012-07-10 16:51:32.0
Modes d'acquisition du patrimoine	Voir	2012-05-10 11:10:42.0
Modes d'acquisition du patrimoine	Voir	2012-05-10 11:13:41.0
Modes d'acquisition du patrimoine	Voir	2012-05-18 10:14:24.0
Modes d'acquisition du patrimoine	Voir	2012-05-27 17:44:52.0
Modes d'acquisition du patrimoine	Voir	2012-07-10 16:53:45.0

FIGURE A.12 – Le détail de l'activité d'un élève en activités SCORM réalisées

Temps total passé en ligne
194 heures 40 minutes et 51 secondes

Sessions de connexion

Début et fin de la session	Durée de la session
lundi 30 avril 2012 21:14:44 - lundi 30 avril 2012 23:39:40	2 heures 24 minutes et 56 secondes
mardi 01 mai 2012 15:45:22 - mardi 01 mai 2012 18:17:42	2 heures 32 minutes et 20 secondes
mardi 01 mai 2012 20:38:22 - mardi 01 mai 2012 23:02:08	2 heures 23 minutes et 46 secondes
mercredi 02 mai 2012 20:56:02 - mercredi 02 mai 2012 23:42:06	2 heures 46 minutes et 4 secondes
jeudi 03 mai 2012 20:30:49 - jeudi 03 mai 2012 23:22:33	2 heures 51 minutes et 44 secondes
dimanche 06 mai 2012 00:17:27 - dimanche 06 mai 2012 02:26:51	2 heures 9 minutes et 24 secondes
lundi 07 mai 2012 12:10:43 - lundi 07 mai 2012 14:10:43	2 heures 0 minutes et 0 secondes
mardi 08 mai 2012 14:08:29 - mardi 08 mai 2012 17:17:25	3 heures 8 minutes et 56 secondes
mercredi 09 mai 2012 21:48:33 - mercredi 09 mai 2012 23:54:22	2 heures 5 minutes et 49 secondes
jeudi 10 mai 2012 21:59:56 - vendredi 11 mai 2012 00:05:19	2 heures 5 minutes et 23 secondes
dimanche 13 mai 2012 19:34:45 - dimanche 13 mai 2012 21:41:37	2 heures 6 minutes et 52 secondes
jeudi 24 mai 2012 19:29:46 - jeudi 24 mai 2012 19:31:59	0 heures 2 minutes et 13 secondes
jeudi 24 mai 2012 19:36:22 - jeudi 24 mai 2012 21:41:40	2 heures 5 minutes et 18 secondes

FIGURE A.13 – Le détail de l'activité de connexion d'un élève

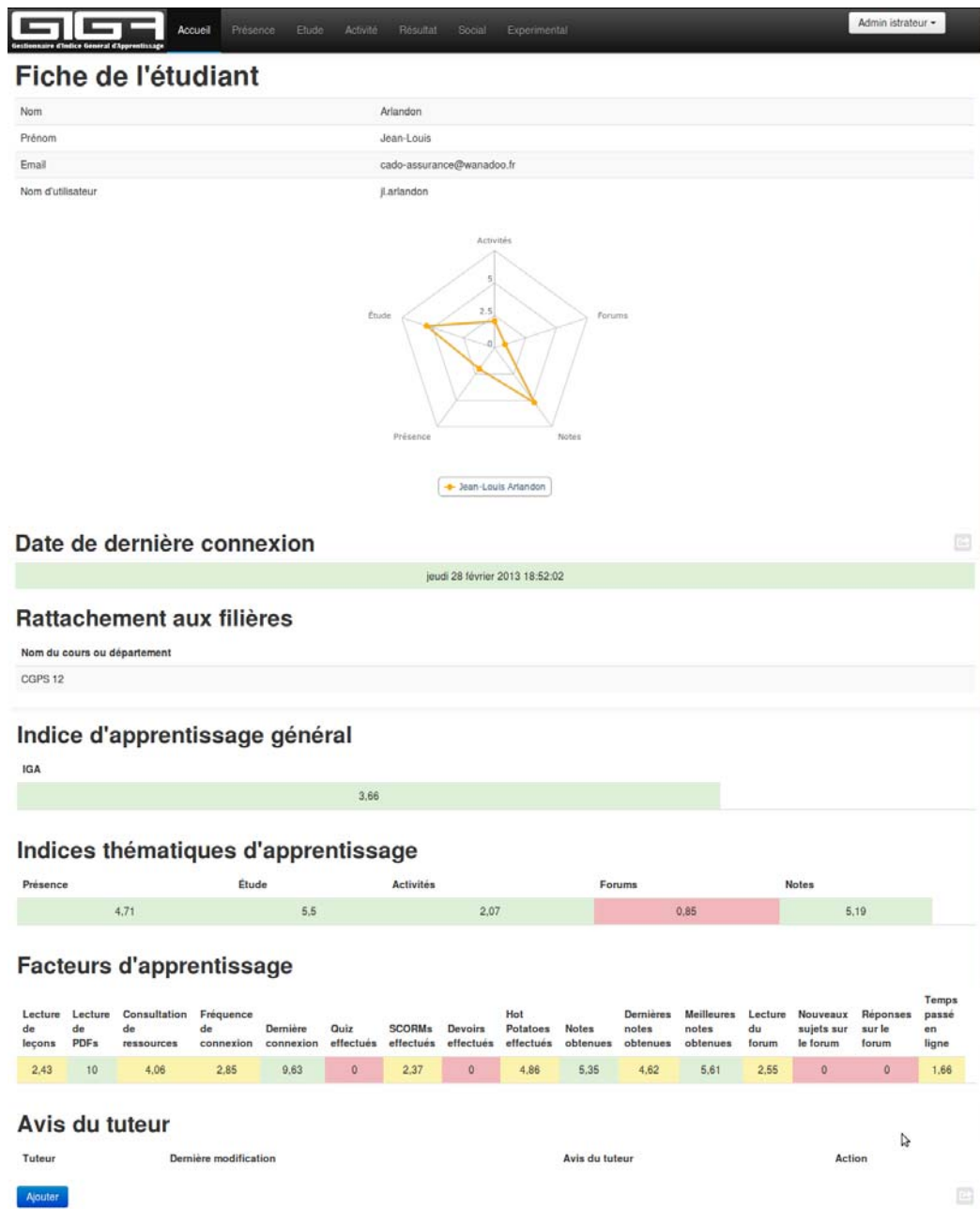


FIGURE A.14 – Le profil d'un élève, regroupant tous ses indicateurs

Annexe B

Diagrammes de classes

Nous présentons dans ce chapitre divers diagrammes de classes associés à l'implémentation de nos propositions.

B.1 HTM-CLA

B.1.1 Implémentation du cœur HTM

Nous présentons dans la figure B.1 le diagramme de classes auquel nous a mené notre conception de l'implémentation du cœur de HTM-CLA. Nous y reconnaissons les grands concepts présentés dans la description de HTM-CLA : une hiérarchie de régions, elles-mêmes composées de colonnes qui contiennent plusieurs cellules ; les segments dendritiques proximaux regroupent au sein d'une colonne les différentes synapses proximales qui proviennent des colonnes de la région inférieure, et les segments dendritiques distaux regroupent au sein des cellules les synapses distales qui proviennent des autres cellules de la même région. Le tout est relié à des entrées réelles via l'interface `Input`, à qui on peut demander de passer au pas de temps suivant, et donc à l'entrée suivante, et qui sera à réaliser dans des classes spécifiques adaptées au type d'entrée.

Nous avons ressenti le besoin de mettre quelques fonctionnalités en commun dans les classes abstraites `Synapse` et `DendriteSegment`. Nous avons également créé l'interface `RegionOrInput`, afin d'exprimer l'entrée à une région donnée, qui peut être soit une entrée (`Input`), soit une région inférieure (`Region`) ; associée à cette interface, un de ses éléments, `RegionOrInputElt`, qui possède des coordonnées et dont on peut consulter l'activation. Nous avons également une classe `Simulation`, qui permet de contrôler la coordination entre la hiérarchie de régions et les entrées. Notons qu'avec les choix d'implémentation actuels, c'est donc la simulation qui demande aux entrées leur prochaine valeur, qui doit être disponible ; on pourrait imaginer un autre modèle dans lequel ce sont les entrées qui font remonter l'information qu'elles ont une nouvelle valeur disponible, mais nous n'avons pas fait ce choix, puisqu'il ne correspondait pas à nos besoins.

L'interface `Input` doit être réalisée d'une façon qui permet de convertir un type de

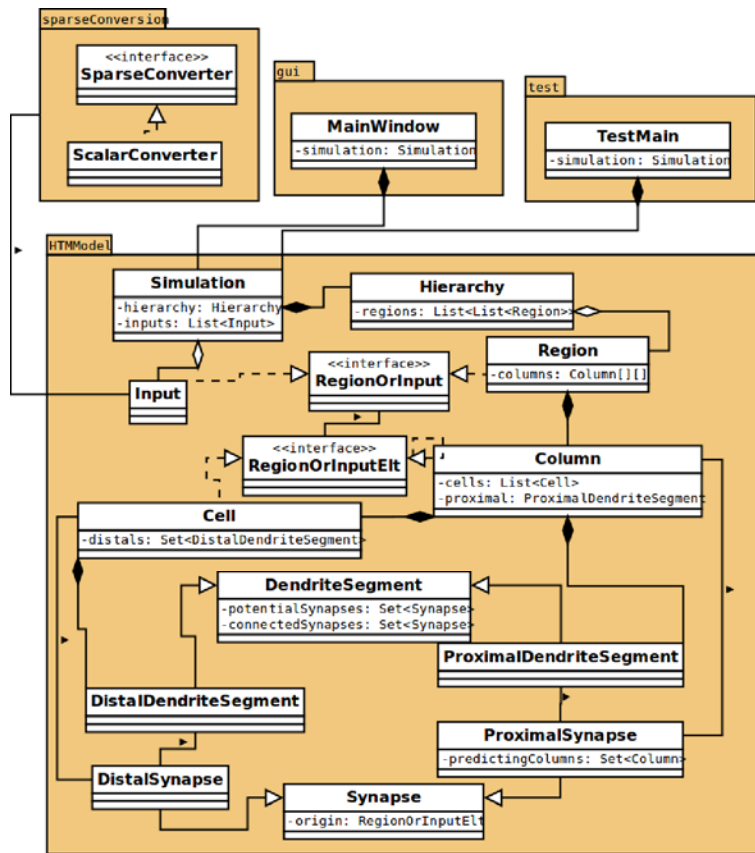


FIGURE B.1 – Diagramme de classes simplifié de HTM-CLA

données donné en une représentation booléenne clairsemée. Cette conversion dépend du type de données, ce qui explique que nous n’ayons pas proposé de réalisation de cette interface. On pourrait toutefois, dans un futur proche, imaginer de proposer une méthode de conversion des types les plus usuels (nombres entiers, nombres flottants...)

Nous avons également proposé une implémentation correspondant à la contrainte de la représentation clairsemée. En effet, nous avons implémenté des régions génériques, qui doivent pouvoir fonctionner de la même façon, quelle qu’en soit l’entrée, soit une entrée de base (Input), soit une autre région. Rappelons que c’est le rôle de l’interface RegionOrInput que de permettre cette genericité. Mais pour que l’entrée se comporte comme une région, elle doit respecter deux contraintes : être représentée par un ensemble d’unités binaires, et par une représentation clairsemée au sein de ces unités.

Pour cela, nous avons créé un paquet intitulé sparseConversion, qui contient une interface SparseConverter, que toutes les classes de convertisseur doivent réaliser. La conversion se fait à l’intérieur d’une classe d’entrée qui réalisera l’interface Input. Pour l’instant, nous avons proposé une classe intitulée ScalarConverter qui réaliser

l'interface SparseConverter et sert à convertir des nombres.

Le deuxième diagramme B.2 détaille les méthodes disponibles dans chaque classe, afin de détailler le fonctionnement de l'ensemble.

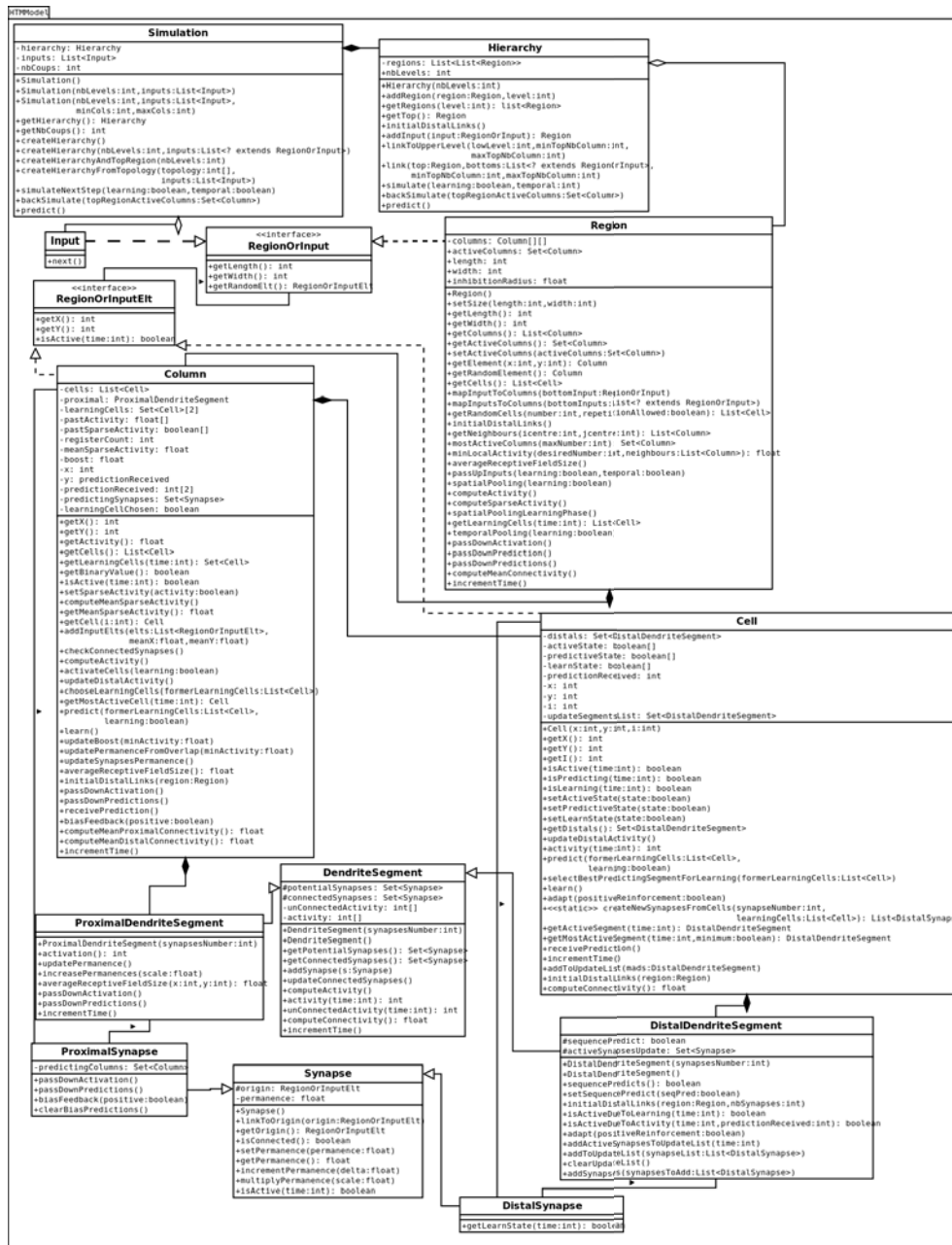


FIGURE B.2 – Diagramme de classes détaillé de HTM-CLA

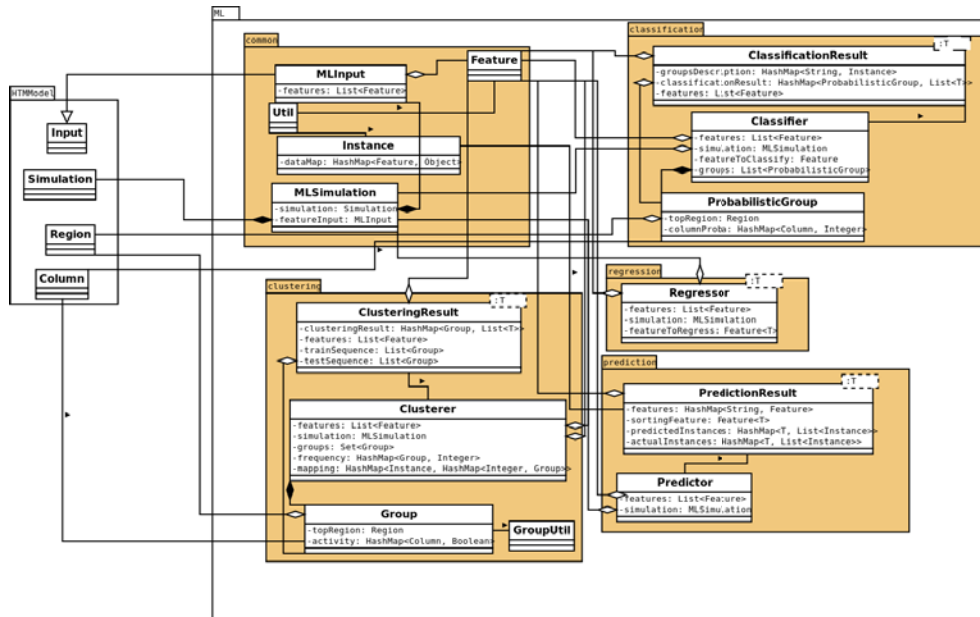


FIGURE B.3 – Diagramme simplifié des classes permettant les fonctionnalités d'apprentissage automatique

B.1.2 Implémentation de l'apprentissage automatique

Comme expliqué dans la partie 4.3, nous avons choisi de recréer diverses méthodes d'apprentissage automatique, toutes basées sur l'algorithme HTM-CLA. Il en a résulté un paquet indépendant, qu'on peut qualifier de librairie, que nous avons séparés en plusieurs sous-paquets. Le diagramme de classe correspondant est présenté à la figure B.3. Un second diagramme plus détaillé est présenté à la figure B.4.

Le premier, `common`, contient l'ensemble des concepts nécessaires à l'apprentissage automatique en général. On y retrouve la notion de caractéristique, ou `Feature`, et d'instance, qui est un ensemble de valeurs associées chacune à une caractéristique. Associée à ces concepts est la classe `MLInput`, qui réalise la classe abstraite `Input` mentionnée plus haut et qui fait la conversion des instances en activation d'unités binaires.

Nous avons ensuite un sous-paquet dédié à chaque type de méthode proposée, dans lequel on retrouve généralement une classe dédiée à la réalisation de l'algorithme, une classe dédiée au stockage des résultats, et éventuellement des classes utilitaires.

B.1.3 Génération de comportement avec HTM-CLA

Comme expliqué dans la partie 4.3.5, nous proposons d'utiliser l'algorithme HTM-CLA comme moteur de base d'un système de génération de comportement, qui lui-même servirait de base à la création d'un tuteur virtuel intelligent adapté au suivi des élèves. Nous montrons ici les détails de l'implémentation de notre proposition.

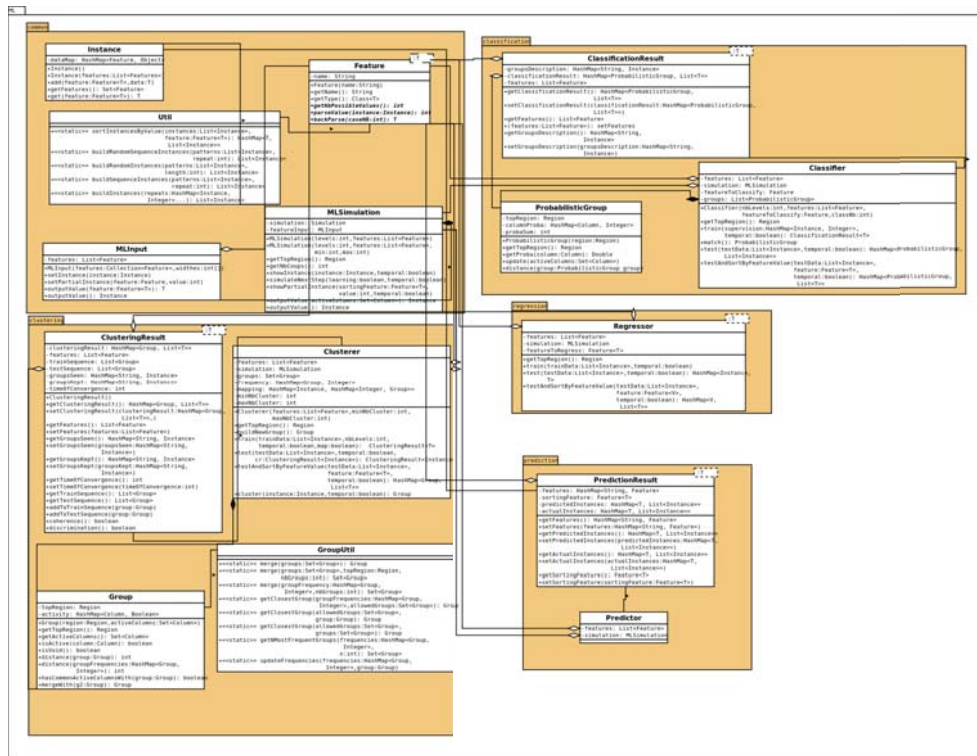


FIGURE B.4 – Diagramme détaillé des classes permettant les fonctionnalités d'apprentissage automatique

Nous proposons à la figure B.5 un modèle qui, comme souhaité, intègre les actions élémentaires en bas de la hiérarchie. Pour cela, nous allons créer une nouvelle région, située en bas de la hiérarchie, dont le seul but sera de modéliser ces actions ; ainsi qu'un convertisseur permettant de transformer nos actions pour servir d'entrée à une région générique, c'est-à-dire un convertisseur en représentation sous forme d'un tableau à deux dimensions d'unités binaires à activation clairsemée. Nous créons donc une interface ActionInput pour représenter cette entrée.

Nous en proposons également une première réalisation baptisée BehaviourInput. Comme nous l'avons suggéré, elle utilise le mécanisme de prédiction comme mécanisme de sélection d'action. À chaque pas de temps, l'action qui sera prédite comme la plus probable sera réellement effectuée. Cette action sélectionnée sera ensuite l'entrée activée au pas de temps suivant. Nous avons également une classe ActionSimulation qui gère la coordination entre BehaviourInput et le reste de la simulation.

Dans chacun des domaines de test utilisés, les actions élémentaires possibles seront modélisées sous la forme d'une énumération ad hoc. On pourra éventuellement rajouter à cette énumération des méthodes si le domaine en fournit l'utilité. C'est pour cela que la plupart des classes rajoutées sont génériques et dépendent du type des actions.

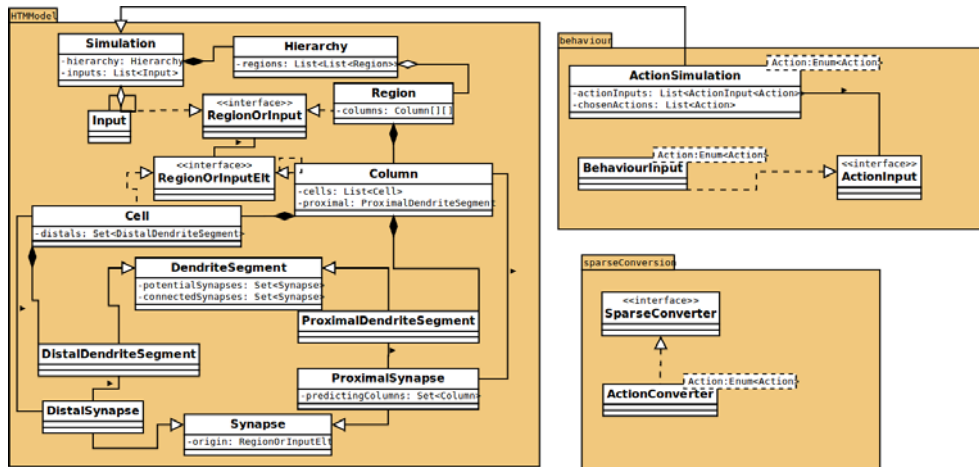


FIGURE B.5 – Le diagramme de classes de l'ajout de comportement à HTM

Pour réaliser la conversion en représentation clairsemée, nous avons prévu une nouvelle classe qui réalise l'interface `SparseConverter`. Elle s'intitulera donc `ActionConverter`. La conversion des actions est plus simple que celle des nombres, puisqu'on ne cherche pas à établir de liens sémantiques entre les actions. Nous n'allons donc pas chercher à produire des recouvrements d'activation d'unités contiguës. La méthode pour calculer le nombre de cases est la même que pour `ScalarConverter` et cherche toujours à créer un taux de clairsemage de 5%.

Nous présentons une première démonstration de l'instanciation sous forme de classes pour nos deux domaines de test, Maze/Woods et pierre-feuille-ciseaux, dans la figure B.6. Nous y remarquons que les classes obtenues sont très similaires dans les deux cas. La seule différence visible au niveau du diagramme de classes est l'utilisation du convertisseur clairsemé, qui n'est pas le même, parce que les perceptions dans les deux problèmes ne sont pas du même type : pour pierre-feuille-ciseaux, la seule information à percevoir est l'action jouée par les deux joueurs, d'où l'utilisation de `ActionConverter` ; pour Maze/Woods, tout l'environnement spatial intervient en plus de l'action déjà prise en compte par l'inclusion de `BehaviourInput`.

B.2 GIGA

Suite au découpage architectural de GIGA, nous avons conçu les diverses classes qui allaient composer cette application, et les paquets qui allaient les regrouper. Nous n'allons pas prétendre à la description exhaustive d'une application de cette taille, mais nous allons toutefois montrer les liens entre les principaux paquets dans le diagramme B.7, et lister dans la figure B.8 les classes contenues dans le paquet `entities`, que nous appellerons entités et qui correspondent aux concepts utilisés dans notre application et aux objets stockés dans la base de données. Nous ne spécifierons pour l'instant pas dans ce deuxième diagramme les entités des paquets `iga` et `ML`, qui correspondent respectivement aux adaptations liées aux indicateurs

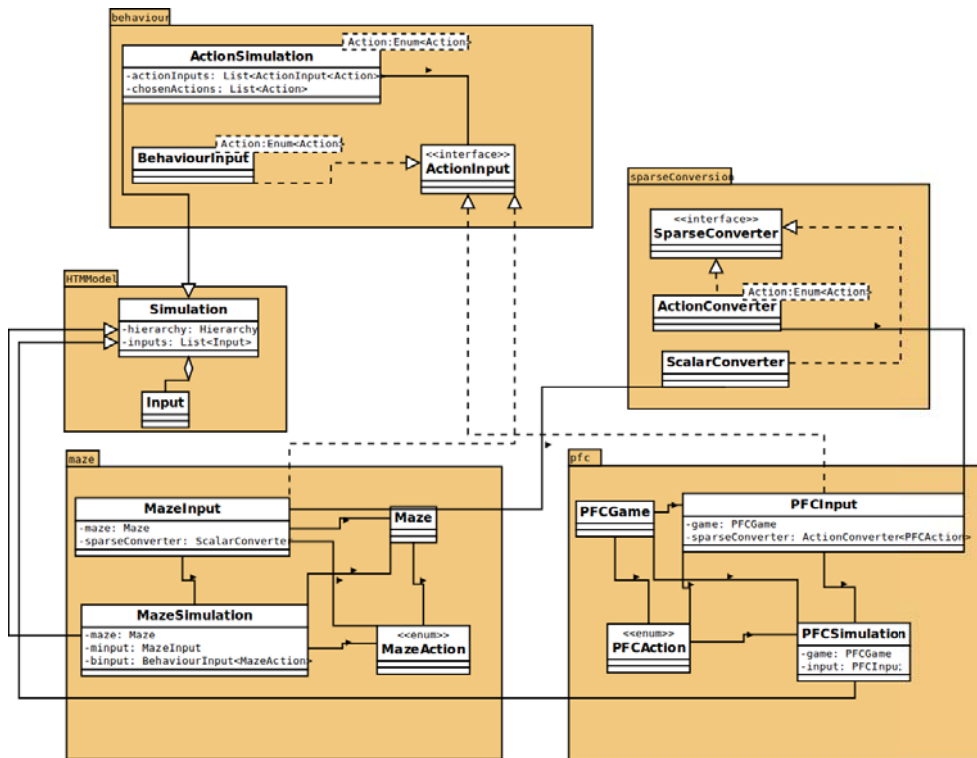


FIGURE B.6 – Le diagramme de classes des implémentations Maze et pierre-feuille-ciseaux

et à l'apprentissage automatique, et qu'on étudiera dans les parties dédiées.

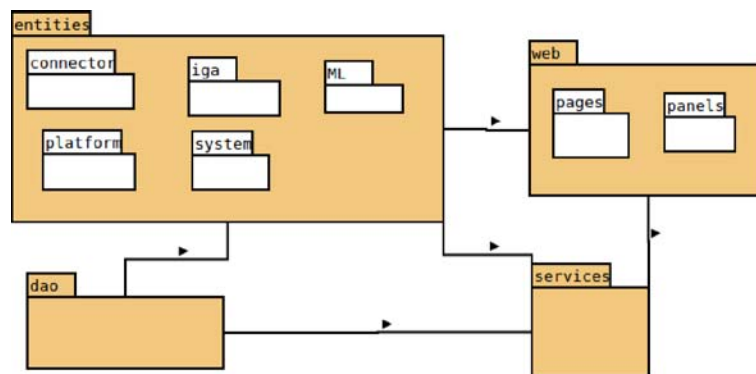


FIGURE B.7 – Les paquets principaux constituant notre application

Comme le premier diagramme le montre, chaque entité est associée à une classe permettant d'accéder aux objets de ce type dans la base de données, située dans le paquet dao. Elle est également associée à une classe permettant de traiter les données de ce type, situé dans le paquet service. Enfin, tout l'aspect affichage est géré par les pages et les panneaux situés dans le paquet web.

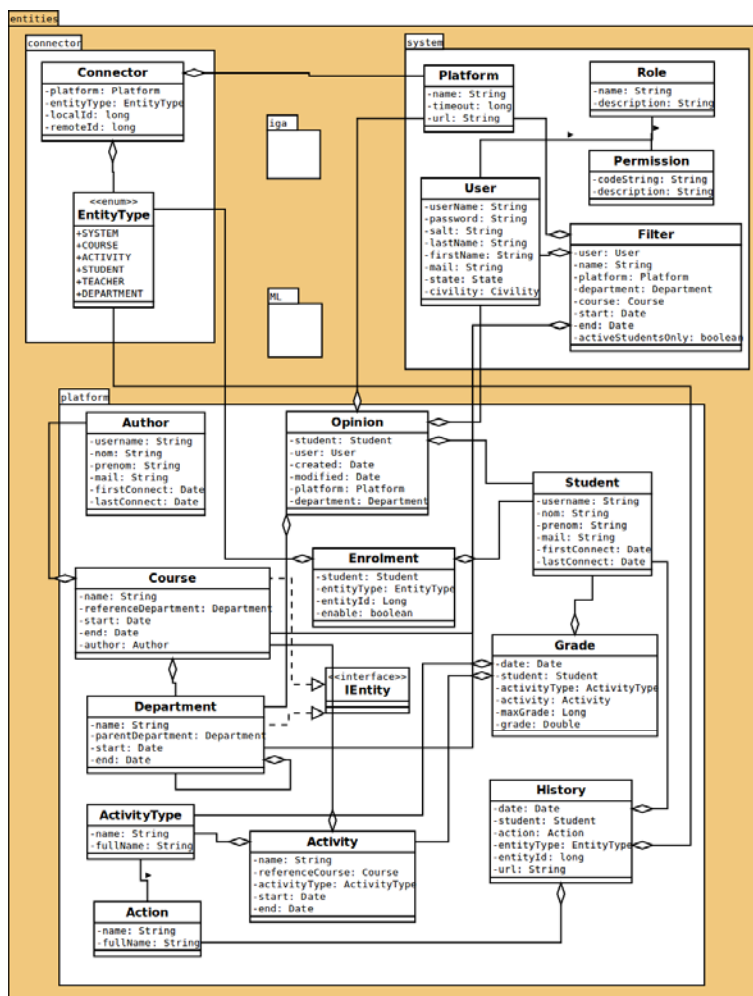


FIGURE B.8 – Le diagramme de classes des entités de GIGA

B.2.1 Indicateurs

Pour les indicateurs proposés à la section 2.2, nous avons proposé un ensemble d'entités nouvelles, et les autres classes associées. Les entités sont présentées dans le diagramme B.9.

Pour rappel, nous avons proposé différents indicateurs, mais également des indicateurs thématiques et un indicateur général, qui correspondent à une moyenne pondérée des indicateurs de niveau hiérarchique inférieur. Nos entités doivent donc refléter aussi bien la possibilité de calculer directement les indicateurs de bas niveau que la structure hiérarchique les reliant à ceux de plus haut niveau. Enfin, nous souhaitons laisser la possibilité à l'utilisateur de personnaliser la pondération.

La classe principale est Factor, qui représente un indicateur, caractérisé principalement par son type, qui correspond à une autre entité, FactorType. Ce type peut prendre les valeurs « données brutes », « données normalisées » ou « composite ».

Lorsqu'il s'agit d'une donnée brute ou d'une donnée normalisée, on stocke alors la méthode pour calculer cet indicateur dans l'attribut `computationMethodName` de `Factor`. Lorsque l'indicateur est composite, on va se servir de la classe `FactorLink`, qui modélise les liens hiérarchique entre indicateurs et en stocke également la pondération choisie par les différents utilisateurs de la plate-forme. Enfin, la classe `ColorCoef` sert à chaque utilisateur à fixer, pour un indicateur donné, les seuils qui feront passer le code couleur de rouge à jaune et de jaune à vert.

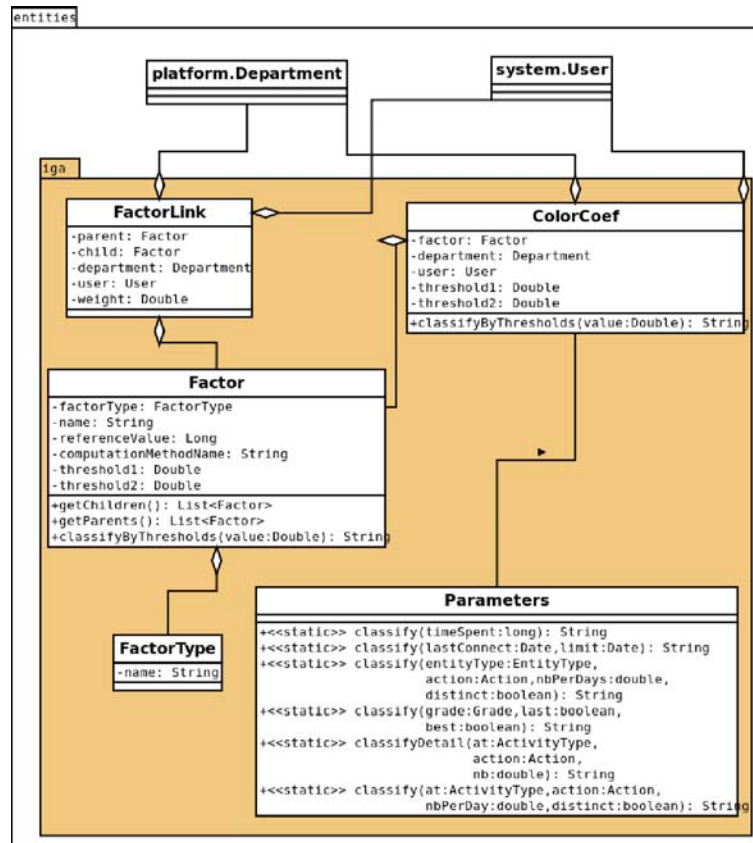


FIGURE B.9 – Le diagramme de classes des entités liées aux indicateurs

B.2.2 Apprentissage automatique

B.2.2.1 Intégration de Weka

Nous avons précisé dans la partie 5.1 notre choix d'utiliser la librairie d'apprentissage automatique Weka. Pour intégrer cette librairie à notre application, nous avons dû réaliser un certain nombre de classes. Certaines, comme `WekaClassificationResult` et `WekaClusteringResult`, nous servent à stocker et encapsuler les résultats fournis par les algorithmes de Weka. Nous avons également créé un nouveau service dont le rôle est de faire appel aux algorithmes de Weka et d'en renvoyer les résultats.

Finalement, un certain nombre de classes sont dédiées à l’affichage de résultats. Nous avons notamment prévu de pouvoir comparer les résultats des algorithmes entre eux.

Le diagramme de classes qui en résulte est présenté à la figure B.10. Le contenu des nouvelles classes est un peu détaillé à la figure B.11.

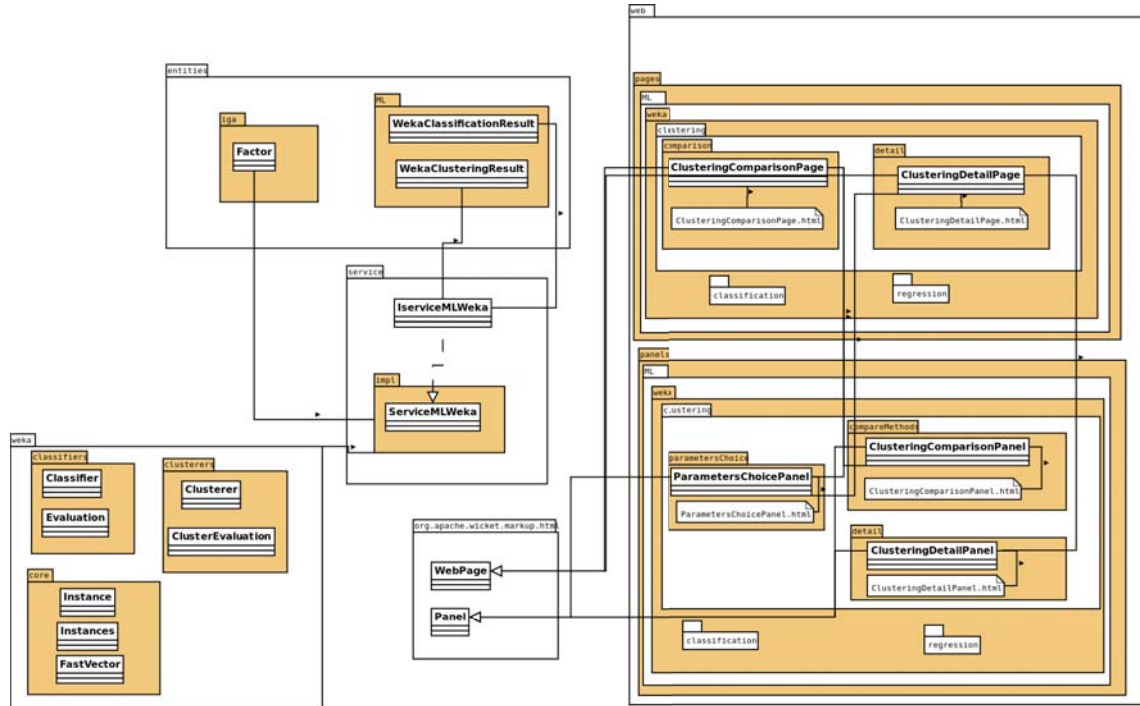


FIGURE B.10 – Le diagramme de classes simplifié de l’intégration de Weka à GIGA

B.2.2.2 Intégration de HTM-CLA

Pour l’intégration de notre librairie HTM à GIGA, il a fallu créer une classe de type service, intitulée ServiceHTM, qui réalise l’ensemble des calculs nécessaires sous forme de lancement des classes Clusterer, Classifier ou Regressor de notre librairie HTM-ML, ainsi que plusieurs classes pour l’affichage des résultats de ces calculs.

Les classes qui en résultent sont présentées sur le diagramme de classe suivant B.12, où seules certaines sont détaillées du fait de leur structure très similaire. Les méthodes utilisées dans le service sont présentées sur le diagramme B.13.

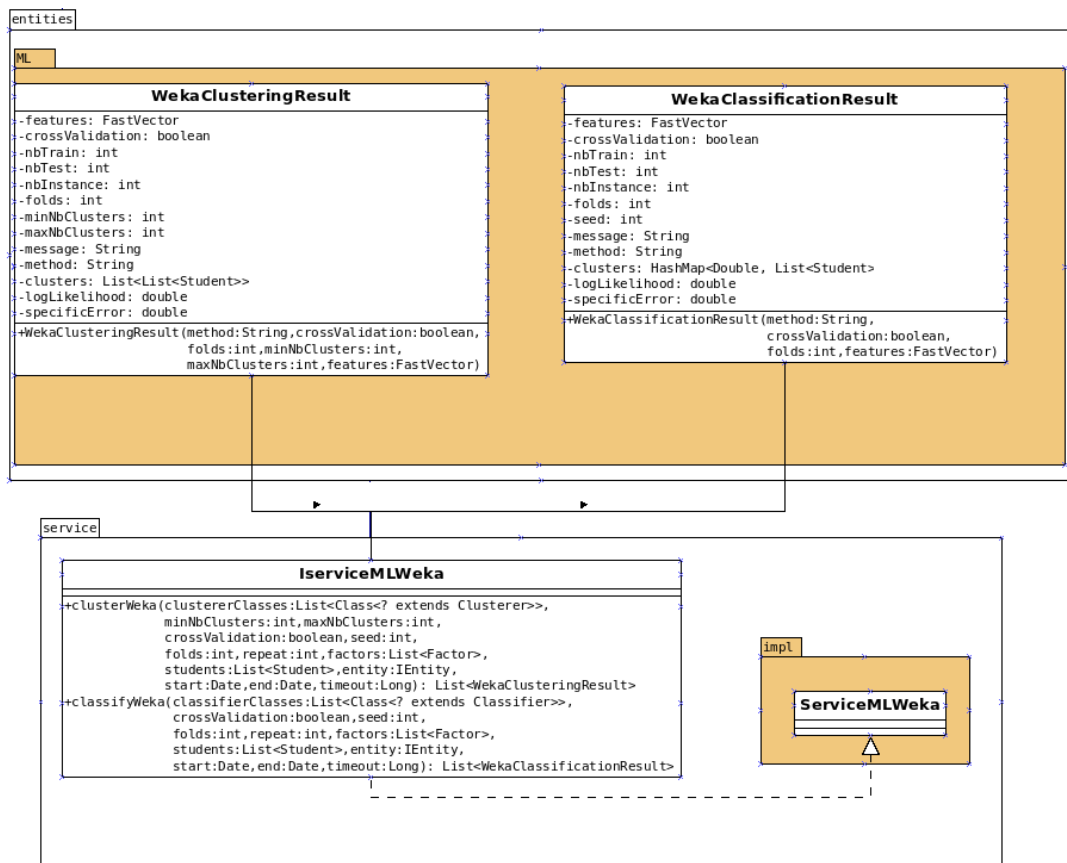


FIGURE B.11 – Le diagramme de classes détaillé de l'intégration de Weka à GIGA

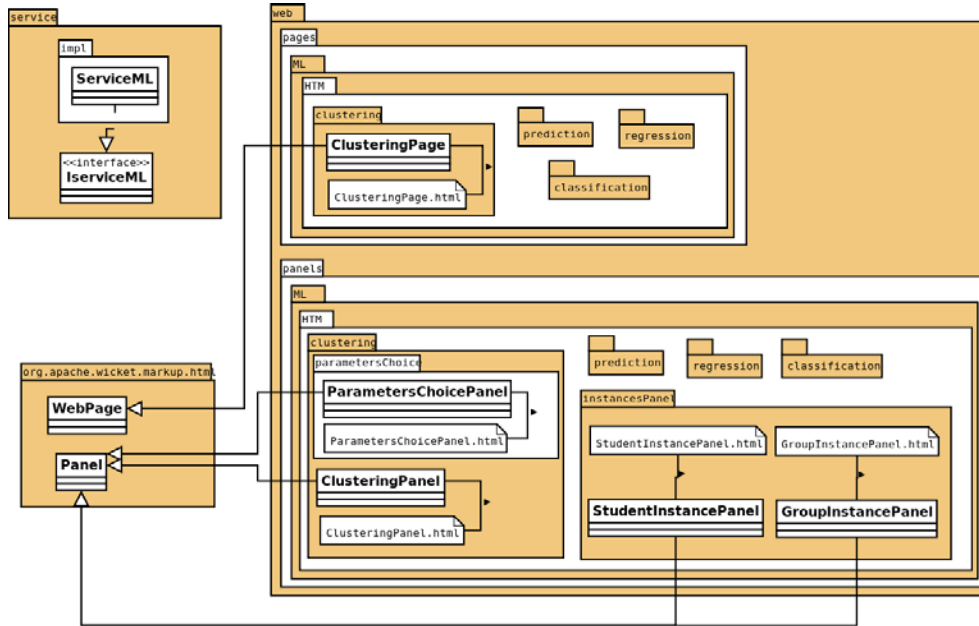


FIGURE B.12 – Diagramme de classes simplifié d'intégration de HTM-CLA à GIGA

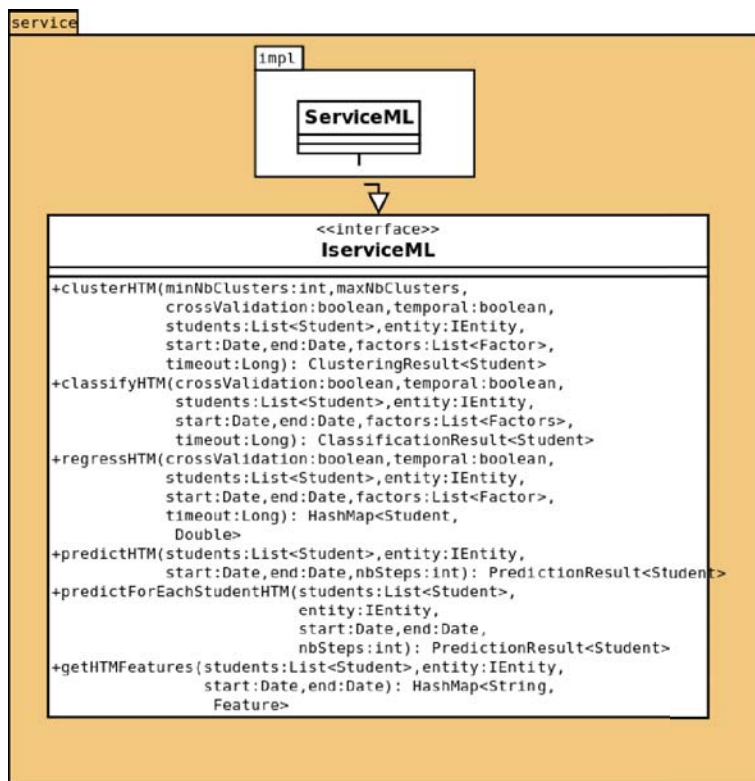


FIGURE B.13 – Diagramme de classes détaillé d'intégration de HTM-CLA à GIGA

Annexe C

Fonctionnement du cerveau humain

C.1 Le cerveau humain

Le cerveau est une partie, certes remarquable, du système nerveux humain. Comme son nom l'indique, le système nerveux est constitué de cellules nerveuses, plus connues sous le nom de neurones, qui sont capables de faire transiter de l'information sous la forme d'un signal électrique ou chimique. Leur fonctionnement sera détaillé plus bas.

Le système nerveux a plusieurs rôles : il reçoit et traite les perceptions sensorielles, coordonne les actions motrices et régule le fonctionnement des autres organes via le système hormonal. Chez les animaux qui ont ces capacités, il est également le siège des émotions et de l'intellect.

Le système nerveux des vertébrés est classiquement décomposé en le système nerveux central, qui contient l'encéphale (cerveau, cervelet et tronc cérébral) et la moelle épinière, et le système nerveux périphérique, qui est constitué des neurones qui traversent tout le corps pour le relier au système nerveux central. Les corps cellulaires de ces neurones périphériques constituent les ganglions nerveux, et leurs longues terminaisons, nos nerfs, font transiter l'information dans les deux sens : la perception vient du corps et remonte vers l'encéphale, qui envoie dans l'autre sens des ordres moteurs. L'encéphale est donc le centre de traitement de l'information nerveuse.

Au sein de l'encéphale, le cervelet permet la coordination et la précision des mouvements du corps, sans en être l'origine ; le tronc cérébral permet le contrôle de l'activité respiratoire et cardiaque, de la pression artérielle, du sommeil et de l'attention ; enfin, le cerveau est le siège de la conscience, des émotions, de l'intellect et du comportement.

En moyenne, la taille du cerveau augmente avec la taille de l'organisme, suivant une loi de puissance, au moins chez les mammifères. Toutefois, les primates ont un cerveau cinq à dix fois plus gros que ce qu'indique cette formule. L'homme en particulier a un cerveau encore plus développé, puisque le rapport entre sa masse

(1,5 kg en moyenne) et celle du corps est sept fois supérieure à celui de la moyenne des mammifères. Le volume cérébral de l'homme a beaucoup augmenté au cours de son évolution, partant de 410 cm³ chez l'australopithèque, le premier hominidé, à 1400 cm³ en moyenne pour homo sapiens.

Le tissu cérébral est composé de 170 milliards de cellules réparties en deux types, les neurones (86 milliards) et les cellules gliales (85 milliards) [Azevedo et al., 2009]. Cependant, seuls les neurones traitent directement l'information nerveuse ; les cellules gliales, ou cellules de soutien, assurent diverses fonctions annexes dont le métabolisme cérébral. Les axones, ou terminaisons des neurones, occupent la majeure partie de l'espace cérébral, à cause de leur très grande longueur, estimée au total à 100 000 km.

Les cellules gliales jouent un rôle de soutien des neurones. On compte parmi elles :

- les astrocytes, qui, dans leur rôle de soutien, forment une charpente structurant les neurones, les nourrissent grâce à leurs liens avec le système sanguin et assurent la défense immunitaire. Néanmoins, ils contribuent aussi à l'activité neuronale : ils recyclent les neurotransmetteurs et contribuent à sculpter le cerveau en contribuant à la neurogenèse et en élaguant certaines synapses.
- les oligodendrocytes, qui gaignent les axones de myéline, substance isolante et protectrice qui augmente la vitesse de l'influx nerveux, dans le système nerveux central (les cellules de Schwann tiennent ce même rôle dans le système nerveux périphérique)
- les cellules microgliales, qui ont un rôle immunitaire, puisque ce sont en fait des macrophages
- les épendymocytes, qui synthétisent le liquide céphalo-rachidien, qui protège le cerveau des chocs.

Dans l'encéphale, les neurones sont situés en périphérie, et les axones au milieu. Cette séparation physique a été remarquée très tôt parce qu'elle induisait une différence de coloration des tissus. En effet, les neurones et les cellules gliales ont une teinte grisâtre, tandis que les axones recouverts de myéline sont plus clairs. C'est ce qui a donné le nom de la « matière grise » et de la « matière blanche ». De ce fait, on dit que la substance grise définit un cortex ("écorce" en grec ancien). Chez l'homme, l'épaisseur du cortex est comprise entre 1 et 4,5 millimètres et sa surface avoisine 2 000 centimètres carré. Dans ce cortex, les cellules gliales sont quatre fois plus nombreuses que les neurones.

Le cerveau humain est composé de deux hémisphères, gauche et droit, selon l'axe de symétrie du corps humain. Les deux hémisphères sont relativement peu interconnectés ; toutes les liaisons doivent passer par une partie du cerveau appelé le corps calleux. Les fonctionnalités de ces hémisphères, en revanche, ne sont pas symétriques. Par exemple, le langage et le calcul sont principalement traités dans l'hémisphère gauche, et les aptitudes visuo-spatiales dans l'hémisphère droit, du moins chez environ 90% de la population. Cette spécialisation asymétrique aurait pu fournir un avantage évolutionnaire en accélérant le traitement des informations, dont la rapidité dépend de la distance de transmission des signaux.

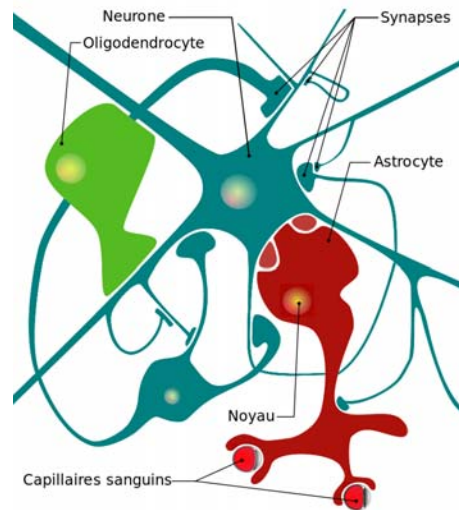


FIGURE C.1 – Cellules gliales

Anatomiquement, on décompose également le cerveau humain en quatre lobes externes : le lobe frontal (situé juste derrière le front), le lobe pariétal (au sommet du crâne), le lobe occipital (à l'arrière du crâne) et le lobe temporal (qui va des tempes vers le dessus des oreilles). C'est également le corps calleux qui fait transiter l'information entre ces lobes.

Le cerveau humain peut également être perçu, selon la théorie du cerveau triunique de Paul McLean [McLean, 1990], comme une superposition de trois couches apparues au fil de l'évolution des espèces. En effet, on peut distinguer de nombreux points communs entre certaines structures de notre cerveau et des structures très similaires chez les autres animaux, notamment vertébrés.

- Le cerveau reptilien ou archécortex est le premier cortex à apparaître, comme son nom ne l'indique pas, chez les poissons lorsque ceux-ci envahissent la terre et deviennent des batraciens. Il apporte les premiers instincts : instinct de conservation, instinct de territoire, agressivité, etc. Il est totalement insensible à l'apprentissage et agit selon des schémas rigides et stéréotypés. Les reptiles, mais aussi les poissons et certains vertébrés inférieurs ne possèdent qu'un cerveau reptilien. Anatomiquement parlant, le cerveau reptilien correspond au tronc cérébral et aux ganglions de la base, à savoir le striatum, le pallidum et le noyau sous-thalamique (voir figure C.3). Ces ganglions servent principalement à inhiber ou désinhiber le thalamus.
- Le cerveau limbique, ou palécortex, apparaît progressivement chez les mammifères. Il leur apporte plusieurs systèmes spécialisés : un contrôle des émotions — affectivité, peur, désir, etc. — et un centre des motivations qui introduit les notions de plaisir, de réussite ou d'échec. En outre, le cerveau limbique permet enfin l'apprentissage. Il ne s'agit que de mettre en relation des situations ou

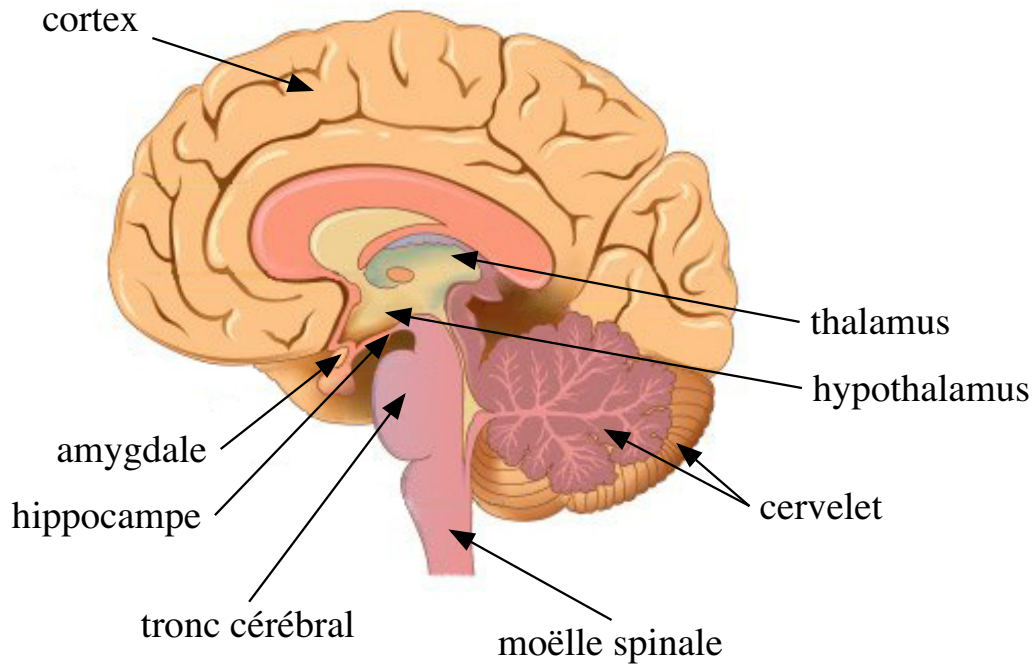


FIGURE C.2 – Vue en coupe des principales structures cérébrales

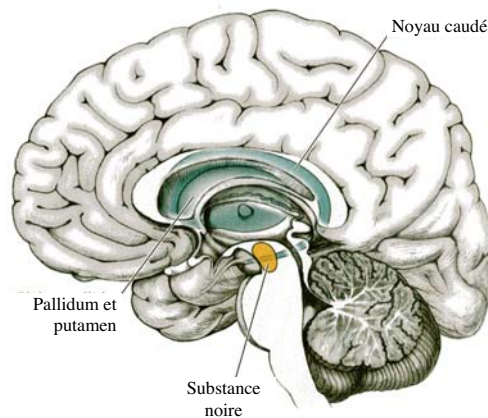


FIGURE C.3 – Ganglions de la base

des actions contextualisées à des sensations, mais cela permet déjà de favoriser l'adaptation de l'individu à son environnement et à un groupe social. Il s'agit d'un mécanisme semblable à celui de l'apprentissage par renforcement, décrit plus haut.

Le cerveau limbique regroupe de nombreuses structures subcorticales, dont

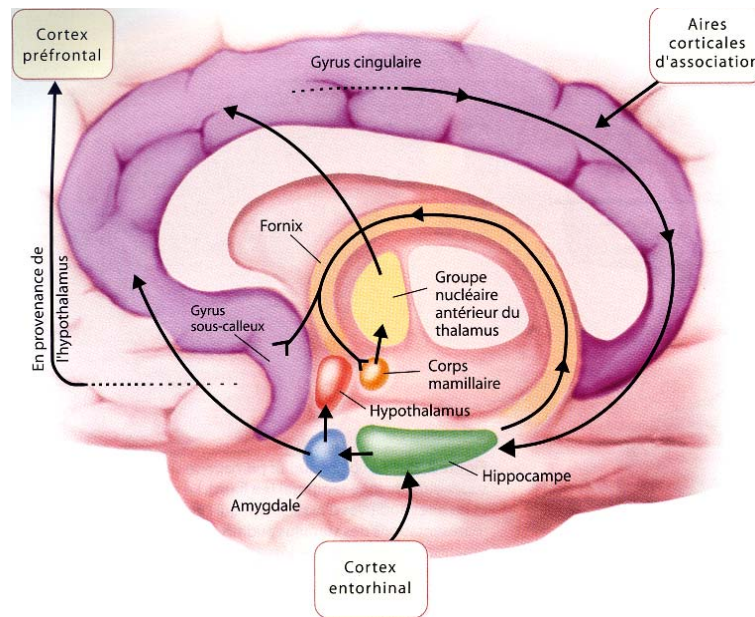


FIGURE C.4 – Cerveau limbique

les trois principales sont l'hippocampe, l'hypothalamus et l'amygdale (voir la figure C.4).

- Le néocortex est la couche la plus phylogéniquement récente du cerveau. Il apparaît chez les grands mammifères, prend de l'importance chez les primates pour finalement culminer avec le genre humain. Au fil de l'évolution, la taille du néocortex humain a considérablement augmenté, jusqu'à représenter maintenant plus de 80% du cortex en termes de nombre de neurones, et cette augmentation a toujours reflété une sensible évolution de ses capacités. Le néocortex est responsable de l'imagination, de la conscience, de la pensée abstraite et de la capacité symbolique.

Aussi bien au regard de ses fonctionnalités que de ce critère de taille, il est donc raisonnable de penser que c'est dans le néocortex que se joue l'essentiel des capacités d'apprentissage de l'humain.

Dans la partie suivante, nous nous attacherons à détailler l'anatomie et le fonctionnement du néocortex.

C.2 Le néocortex

Le néocortex est non seulement la partie du cerveau humain apparue la plus récemment à l'échelle de l'évolution, mais également la plus volumineuse, avec plus de 80% du volume cortical. Pour cette raison, plusieurs scientifiques pensent que c'est dans cette région du cerveau que se situe le siège de l'intelligence spécifiquement humaine. Anatomiquement, il s'agit d'une couche fine située à l'extérieur du cerveau, extrêmement plissée de façon à augmenter sa surface dans un volume restreint. Son

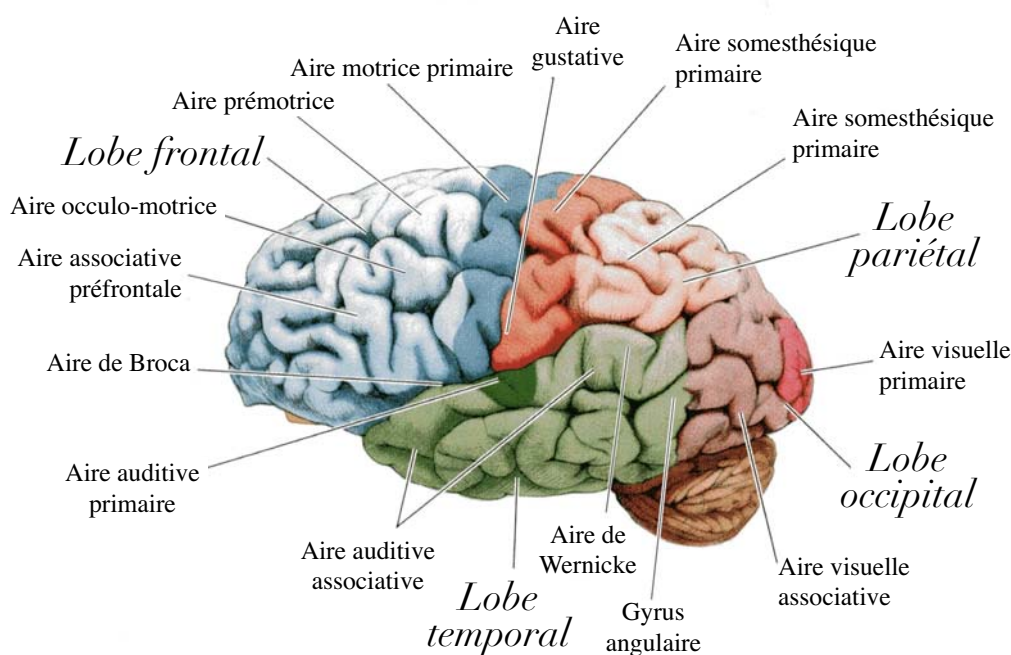


FIGURE C.5 – Les principales aires corticales

épaisseur est en moyenne de 3 millimètres, pour une superficie « dépliée » de 2,2 mètres carrés. Le néocortex contient essentiellement des neurones et des cellules gliales. Il contient deux types principaux de neurones :

- Des neurones pyramidaux qui sont excitateurs (environ 80 % des neurones néocorticaux)
- Des interneurones qui sont inhibiteurs (environ 20 %)

Classiquement, on divise le néocortex en zones fonctionnelles : on retrouve en effet souvent d'un humain à l'autre, au même endroit, des zones dédiées à l'acquisition des sens (vision, audition...), à la motricité, ou encore à l'association entre différents sens. Ces zones sont présentées dans la figure C.5.

Il peut être perturbant de voir plusieurs aires apparemment dédiées à la même fonctionnalité. Quelle différence y a-t-il réellement entre l'aire visuelle primaire et l'aire visuelle associative ? Si l'aire de Broca et l'aire de Wernicke sont toutes les deux cruciales aux capacités du langage, sont-elles redondantes, complémentaires ou des jumelles se divisant la tâche ?

En réalité, chaque aire n'est pas dédiée à un sens ou à un muscle, mais à une partie du traitement général des informations. Pour cela, les neurones de chacune des aires réagissent à différentes propriétés de l'information : ainsi, un neurone va réagir à la couleur bleu, tandis qu'un autre va détecter un mouvement vers la gauche. L'émergence de ces aires s'est en outre accompagnée d'une hiérarchisation, de sorte que le traitement de l'information est assimilable à une chaîne de montage industrielle,

où chaque poste s'appuie sur le travail des postes précédents afin de produire un résultat plus complexe. Les informations sensorielles sont donc progressivement abstraites et complexifiées. C'est ce qui va permettre d'identifier un objet à partir d'un amas de pixels. Réciproquement, on peut présumer que les actions, complexes lorsque le cerveau en prend la décision, sont progressivement décomposées en ordres simples qui seront donnés aux muscles.

Ainsi, certains neurones sont qualifiés de neurones « grand-mère » [Gross, 2002], car ils s'activent en réponse à un signal complexe : un neurone représentant ma grand-mère peut s'activer si je vois ma grand-mère, si je vois une photo de ma grand-mère, si je pense à ma grand-mère, si ma grand-mère me parle au téléphone, si on me parle de ma grand-mère, si j'entends son nom, si je lis une lettre écrite de sa main ou même si quelque chose m'évoque la texture ridée de la peau de ma grand-mère.

Encore plus abstrait, il existe des neurones « miroir » [Rizzolatti and Craighero, 2004], qui s'activent aussi bien quand leur propriétaire effectue une action que quand il observe un autre individu effectuer cette même action, voire même lorsqu'il imagine cette action. Ces neurones ont été proposés comme base physiologique du mécanisme d'empathie.

Précisons toutefois que ni les neurones grand-mère ni les neurones miroir ne sont biologiquement différents des autres neurones : ils doivent probablement leur particularité uniquement à leur câblage au sein du cerveau, qui les relie à d'autres concepts juste en dessous dans la hiérarchie de l'abstraction.

La taille dédiée à chacune des aires sensorielles comme motrices, et donc le nombre de neurones, est proportionnelle à la « sensibilité » accordée à ce sens, ou à la dextérité d'une fonction motrice. Cette notion est schématisée sous la forme d'un homoncule, qui traduit la représentation de l'organisme à l'intérieur du cortex, en accordant à chaque partie une taille proportionnelle à la quantité de tissu cortical qui lui est dédiée. L'homoncule somesthésique (figure C.6A) traduit la sensibilité de la perception de chaque partie de notre corps. L'homoncule moteur (figure C.6B) représente l'organisation de la commande motrice des muscles de notre corps.

Il ne faut pas oublier que cette répartition en zones fonctionnelles ne se vérifie que statistiquement et pas universellement. Ainsi par exemple, le cerveau des gauchers est souvent différent de celui des droitiers. Environ 10% de la population ne présente pas une forte prédominance de l'utilisation de la main droite ; cette catégorie regroupe les gauchers, ex-gauchers contrariés, ambidextres et droitiers à faible préférence. Cette différence est en fait d'origine génétique et se traduit également par une organisation différente de leur cortex. Notamment, les zones du langage, situées dans l'hémisphère gauche chez les droitiers, peuvent chez les gauchers être entièrement dans l'hémisphère droit ou même réparties entre les deux hémisphères. Les gauchers ont également plus souvent des difficultés avec les tâches à nature spatio-visuelle, comme par exemple la compréhension d'une carte routière, ou le jeu d'échecs. Par ailleurs, si chez presque toute l'humanité, les consonnes sont traitées dans l'hémisphère gauche et les voyelles dans l'hémisphère droit, les travaux de Tadanobu Tsunoda [Tsunoda, 1985] auraient montré que chez les Japonais et certaines tribus polynésiennes, les sons du langage

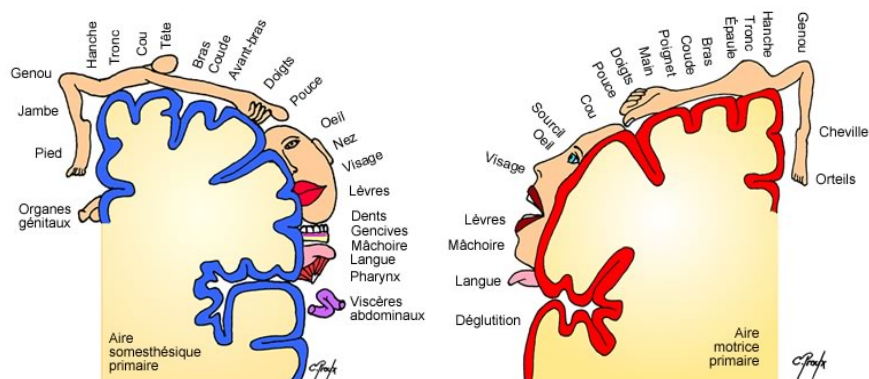


FIGURE C.6 – Les homoncules somesthésique et moteur

sont tous traités dans le même hémisphère, contrairement aux sons mécaniques. Il ne s'agit pas là d'une différence d'origine génétique mais bien due à l'apprentissage.

Il faut également noter que ces différentes zones ne présentent pas de différences structurales : le néocortex est uniforme et il serait impossible de dire, en observant au microscope un morceau de tissu prélevé dans un néocortex, son emplacement d'origine dans le cerveau. C'est ce qui laisse à penser, en plus de l'origine génétique de cette répartition géographique, qu'elle relève plus probablement d'un pré-câblage qu'autre chose. Il faut également noter que le cortex est plastique, ce qui signifie que ces zones peuvent se restructurer en cas de besoin. Ainsi, dans le cas d'une personne aveugle dès la naissance, les zones classiquement dédiées à la vision seront redistribuées à d'autres sens, plutôt que d'être laissées à l'abandon.

C.3 Le neurone

Le neurone est une des cellules les plus importantes du système nerveux humain, et notamment du cerveau. Ce sont les neurones qui assurent l'activité cérébrale telle qu'elle peut être mesurée par un électroencéphalogramme. Cette activité est en fait un signal bioélectrique, appelé « influx nerveux » ou « potentiel d'action » et qui se transmet de neurone en neurone.

Le nombre total de neurones du cerveau humain est estimé à environ 90 milliards. Chaque neurone communique avec les autres par le biais de jonctions appelées synapses. Chaque neurone pouvant former en moyenne dix mille synapses, le nombre de synapses dans le cerveau humain est estimé à environ mille milliards.

Un neurone est composé de trois parties : le corps cellulaire est le cœur de la cellule. Il contient entre autres le noyau cellulaire. L'axone sert à transmettre l'influx nerveux relativement loin dans le système nerveux, de 1 mm à plus d'1 m. Il se termine en arborescence débouchant sur de nombreuses synapses. Enfin, les dendrites [Stuart et al., 2007] (environ 7 000 par neurone) sont beaucoup plus proches du corps cellulaire et beaucoup plus ramifiées. Elles servent à recueillir l'influx nerveux

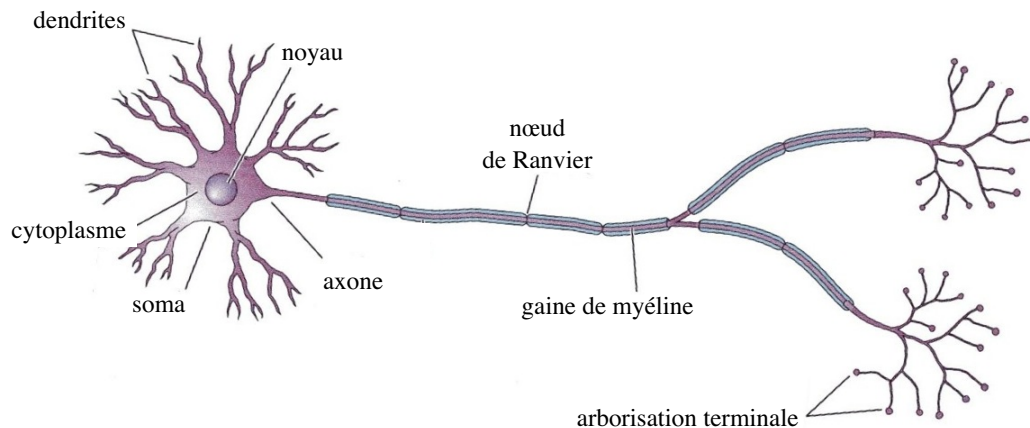


FIGURE C.7 – Schéma d'un neurone

depuis une synapse et le conduire jusqu'au corps cellulaire. Elles possèdent parfois des excroissances appelées épines dendritiques, qui servent à augmenter la surface autour d'une synapse et aider à la transmission du signal.

C.3.1 Potentiel d'action

Selon Bernard Calvino [Calvino, 2006], on peut considérer qu'un neurone fait 5 types d'opérations : l'intégration, le traitement, l'encodage, la genèse et la propagation de l'information nerveuse, depuis ses synapses d'entrées et vers ses synapses de sortie. Le signal part de la membrane post-synaptique (le côté récepteur de la synapse qui a transmis le signal), atteint le corps cellulaire, puis se propage le long de l'axone où il activera d'autres synapses.

L'influx nerveux se traduit physiquement par une modification temporaire de la polarisation électrique d'une partie du neurone, qui consiste en une dépolarisation suivie d'une repolarisation. Cette variation se fait par un processus chimique, l'entrée et la sortie d'ions à travers la membrane cellulaire. C'est la différence de concentration entre les ions sodium (Na^+) et potassium (K^+) entre l'intérieur et l'extérieur de la membrane qui constitue une différence de potentiel entre la face intracellulaire de la membrane du neurone et sa face extracellulaire, appelée potentiel électrique du neurone. Au repos, ce potentiel est d'environ -70 mV.

Localement, lorsqu'une dépolarisation se produit, elle se traduit par l'ouverture de canaux à ions sodium, qui font pénétrer ces ions dans la cellule. Cela modifie leur concentration et provoque donc un saut du potentiel jusqu'à environ 30 mV, valeur proche du potentiel électro-chimique d'équilibre du sodium, pendant 1 ms. Ensuite, des canaux à ions potassium s'ouvrent ; mais ils fonctionnent dans l'autre sens, faisant fuir les ions potassium vers l'extérieur de la cellule. Cette phase de repolarisation dure 1 à 2 ms et amène le potentiel en-dessous de sa valeur de repos : on parle d'hyperpolarisation. Enfin, le potentiel remonte se stabiliser autour de sa valeur de repos, alors que les concentrations en ions se rétablissent à leur valeurs

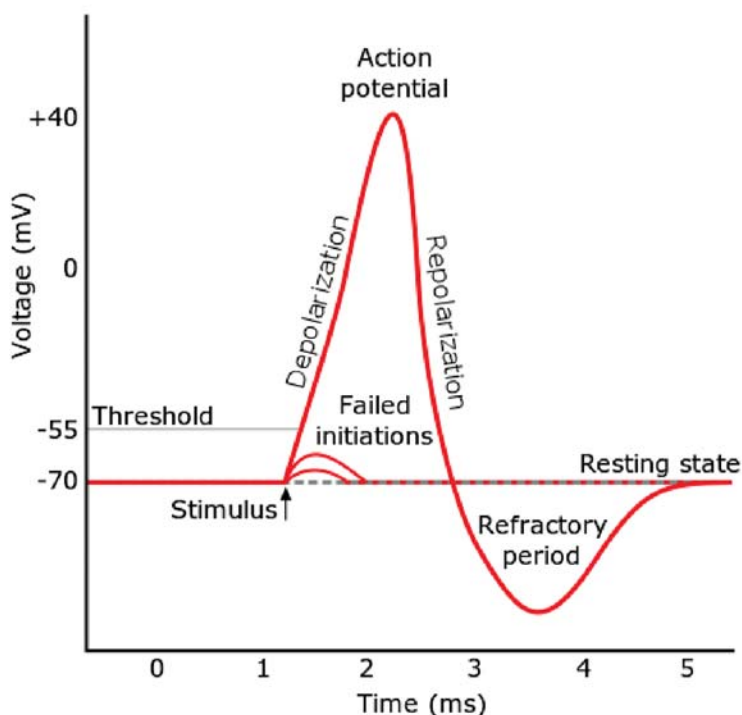


FIGURE C.8 – Potentiel d'action d'un neurone

d'origine.

Suite au déclenchement d'un potentiel d'action, le neurone est temporairement réfractaire à toute stimulation durant la phase d'hyperpolarisation, et ne pourra pas déclencher de nouveau potentiel d'action, par inactivation des canaux sodiques. Cela constitue un temps de latence limitant la fréquence des potentiels d'action. Suite à cette période réfractaire absolue vient une période réfractaire relative, où seules d'importantes stimulations peuvent déclencher un potentiel d'action. Cette dépolarisation se transmet de proche en proche. Cette propagation provient de petits courants électriques très locaux qui s'établissent entre portion excitée et portion non encore excitée. Dans les axones myélinisés, le signal nerveux se propage à grande vitesse (environ 50 m/s chez l'homme), et sans perdre en intensité.

L'origine de ce potentiel d'action provient des membranes post-synaptiques du neurone. Les synapses associées génèrent des potentiels post-synaptiques, qui peuvent être de deux types, excitateur ou inhibiteur. Un potentiel post-synaptique excitateur correspond à une dépolarisation, alors qu'un potentiel post-synaptique inhibiteur est une hyperpolarisation. Ces modifications de la polarisation au niveau post-synaptique se font également par modification de la concentration en ions. Elles se transmettent ensuite le long de la membrane de la dendrite puis du corps cellulaire en s'atténuant peu à peu.

Ces variations de potentiel s'additionnent selon une somme à la fois spatiale (les

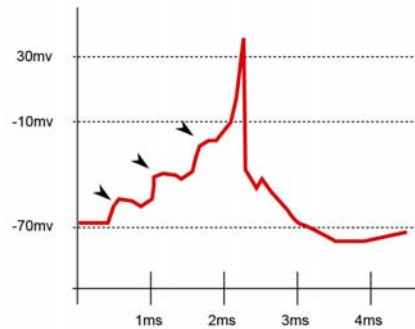


FIGURE C.9 – Potentiels post-synaptiques excitatoires suffisant pour déclencher un potentiel d'action

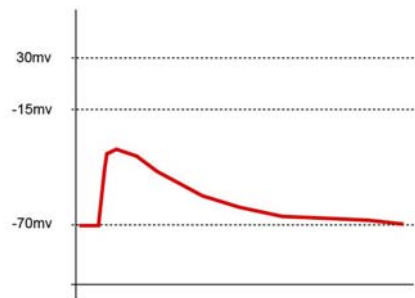


FIGURE C.10 – Potentiel post-synaptique excitatoire insuffisant pour déclencher un potentiel d'action

potentiels post-synaptiques provenant de deux synapses différentes s'additionnent) et temporelle (les potentiels post-synaptiques suffisamment proches dans le temps peuvent s'additionner). Le phénomène d'addition temporelle est dû à la capacité électrique de la membrane, qui prolonge l'effet des potentiels post-synaptiques.

Dans cette addition, les potentiels excitateurs s'opposent aux potentiels inhibiteurs. Si les potentiels excitateurs l'emportent avec une marge suffisante, c'est-à-dire si la dépolarisation totale est supérieure à un certain seuil, alors un potentiel d'action sera généré. C'est la loi du tout ou rien. Ce seuil correspond à la valeur du potentiel de membrane qui active un nombre suffisant de canaux Na^+ , environ -50 mV .

Le déclenchement ou non du potentiel d'action est décidé au niveau du cône d'émergence de l'axone, qui possède davantage de canaux sodiques que les dendrites et le corps cellulaire.

Il est à noter qu'il existe également un phénomène de rétropropagation [Djurisic et al., 2004], où un potentiel d'action se déplace non pas vers la terminaison de l'axone (propagation normale), mais au rebours, en direction des dendrites, d'où provenait la dépolarisation originelle. Toutefois, ce signal perd rapidement en intensité et ne

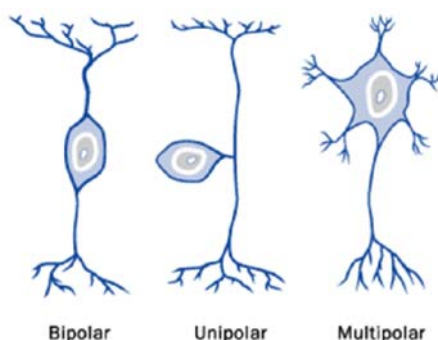


FIGURE C.11 – Polarité des neurones

franchit pas toujours les synapses. Cela remet en question l'idée que les dendrites ne seraient que de simples récepteurs passifs de l'information et les axones les seuls émetteurs. Ces données suggèrent par ailleurs que le neurone n'agit pas seulement de manière élémentaire, mais que des opérations complexes peuvent aussi se produire en son sein.

Un fort stimulus peut mener à une brève période durant laquelle se déclenche une bouffée de potentiels d'action très rapprochés, séparés seulement par la période réfractaire. Ces bouffées seraient au cœur des mécanismes de synchronisation fine.

C.3.2 Types de neurones

On peut définir certaines catégories de neurones spécifiques [Mountcastle, 1998], basées sur des différences fonctionnelles et/ou morphologiques.

Ainsi, on peut fonctionnellement différencier les neurones sensoriels ou afférents, qui sont connectés aux nerfs qui transmettent les informations venant de nos capteurs sensoriels (vision, toucher, etc.), des neurones moteurs ou efférents, qui transmettent des ordres d'actions à nos muscles via d'autres nerfs, que ces ordres soient conscients (déplacement du bras) ou non (battement du cœur). Les autres neurones sont parfois nommés par défaut interneurones.

Du point de vue morphologique, il existe également plusieurs types de neurones. Une première classification en deux types a été proposée par Golgi. Un neurone de type I a un long axone, tandis qu'un neurone de type II a un axone très court, facile à confondre avec une dendrite, voire pas d'axone du tout. En plus de la longueur, les sites d'émergence des axones et dendrites à l'issue du corps cellulaire permettent de proposer une nouvelle classification : les cellules unipolaires ont une seule excroissance d'où partent dendrite et axone, les cellules bipolaires ont deux protubérances partant de côtés opposés du corps cellulaire, l'une dédiée à l'axone, l'autre à une dendrite ; enfin, les cellules multipolaires ont plusieurs dendrites, séparées de l'axone. Enfin, certains neurones dits épineux ont des épines dendritiques, les autres non.

Des types supplémentaires de neurones ont été isolés d'après leur forme et leur localisation dans le cerveau. Les premiers découverts ont été les neurones pyramidaux,

nommés d'après la forme triangulaire, vue en coupe, de leur corps cellulaire. Ils représentent plus de la moitié des neurones corticaux. Ils présentent également la particularité d'avoir deux structures dendritiques différenciées : une très longue dendrite, dite apicale, part du sommet, ou apex, du corps cellulaire ; tandis que plusieurs (3 à 5) dendrites dites basales rayonnent de la base de la pyramide. Ces neurones figurent également parmi les plus gros, avec un corps cellulaire d'environ 20 μm de long. On peut également remarquer que dans les neurones pyramidaux, le nombre d'épines dendritiques augmente avec l'éloignement du corps cellulaire. Ils utilisent généralement le glutamate comme neurotransmetteur excitateur et le GABA comme inhibiteur.

Les dendrites apicales de ces neurones forment des synapses proches du corps cellulaire, dites proximales, mais également des synapses plus éloignées, dites distales [Spruston, 2008]. Selon certaines théories, les neurones pyramidaux s'activent quand des entrées distales coïncident avec des entrées proximales. Une activation proximale faible arrivant légèrement après une activation distale peut également contribuer à amplifier cette dernière. [Larkum et al., 1999, Jarsky et al., 2005]

Une variante géante de ces neurones, les neurones de Betz, sont présents dans le cortex moteur. Ce sont les plus gros neurones, allant jusqu'à 100 μm de diamètre.

Les neurones étoilés ont des dendrites dans toutes les directions, d'où leur nom. Les neurones en paniers sont ainsi nommés d'après la forme de leur arborescence axonale, tout comme les neurones en chandelier, qui ont la spécificité de se relier presque exclusivement au segment initial de l'axone des neurones pyramidaux. C'est en revanche la forme du corps cellulaire qui donne leur nom aux neurones fuseaux, qui sont généralement bipolaires.

Le cervelet contient essentiellement des neurones granulaires et des neurones de Purkinje, ainsi que quelques autres types (cellules de Lugaro, de Golgi et cellules unipolaires à brosse), dont les spécificités ne nous intéressent pas dans le cadre de ce chapitre.

C.4 Synapse

Comme mentionné plus haut, les synapses sont les jonctions entre neurones, et servent à relayer le signal nerveux d'une cellule à l'autre. Il existe deux sortes de synapse :

- Les synapses électriques, assez rares chez les mammifères, où le signal est transmis électriquement par l'intermédiaire d'une jonction communicante (en anglais gap-junction).
- Les synapses chimiques, très majoritaires chez les mammifères et l'homme, qui utilisent certaines molécules, appelées neurotransmetteurs, pour transmettre l'information.

Il reste toutefois un certain nombre de synapses électriques dans le cerveau humain. On peut supposer que leur plus grande rapidité peut être critique en certains endroits. Cela aurait pour effet de faire fonctionner les neurones ainsi connectés

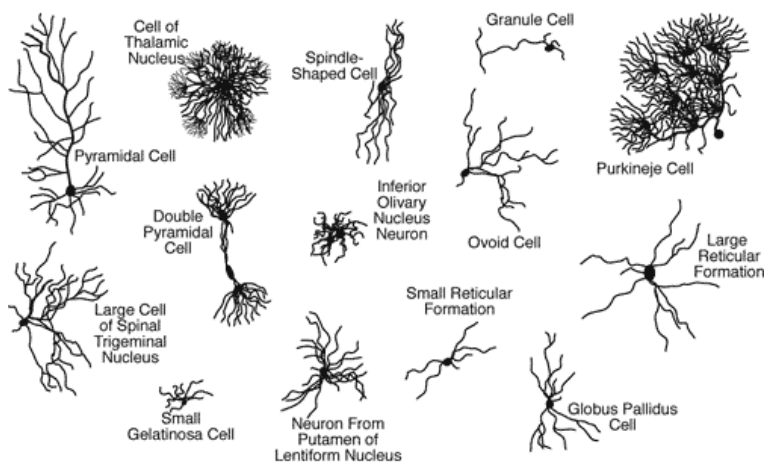


FIGURE C.12 – Types de neurones

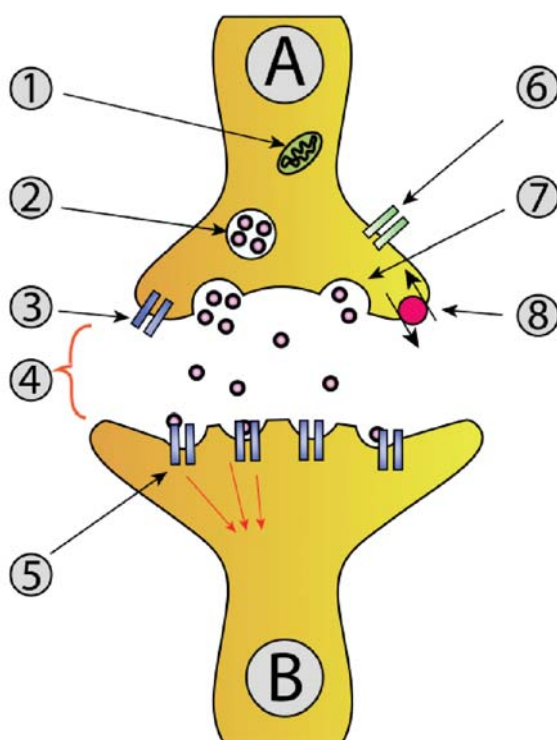


FIGURE C.13 – Synapse chimique

comme un grand ensemble.

Une synapse chimique est constituée de trois parties (on parle de neurones pré-synaptique pour le neurone duquel provient l'influx nerveux et de neurone post-synaptique pour celui vers lequel il est transmis) : L'élément pré-synaptique

est l'endroit où arrive l'influx nerveux. Il s'agit soit de la membrane du bouton terminal de l'axone, soit de la membrane d'une dendrite. C'est là qu'est synthétisé et accumulé le neurotransmetteur, dans des vésicules pré-synaptiques. Quand le potentiel d'action arrive, le neurotransmetteur est libéré. La fente synaptique est l'espace entre les deux cellules, qui mesure environ 20 nm de large. C'est là qu'est libéré le neurotransmetteur. L'élément post-synaptique est celui qui recueille le neurotransmetteur dans des récepteurs post-synaptiques et le convertit en influx nerveux. Il peut être la membrane d'un axone ou d'une dendrite.

On peut repérer au microscope le sens de propagation de l'information car la membrane post-synaptique est beaucoup plus large et dense.

L'arrivée d'un potentiel d'action au niveau d'une synapse déclenche l'ouverture de canaux à ions calcium, faisant entrer ces ions dans la cellule pré-synaptique. Cela provoque la fusion de la membrane des vésicules contenant les neurotransmetteurs avec la membrane plasmique et la libération des neurotransmetteurs. Ce phénomène s'appelle l'exocytose. Il cesse dès l'ouverture de canaux potassium, qui ramène le potentiel de membrane à sa valeur d'origine et inhibe ainsi les canaux calcium.

Certaines synapses jouent un rôle d'excitation, d'autres un rôle d'inhibition de l'influx nerveux. Cela est principalement dû au type de neurotransmetteur utilisé dans la synapse. Ainsi, le glutamate est généralement excitateur, contrairement au GABA. En plus de cette inversion qualitative, un même signal pré-synaptique peut provoquer une réaction plus ou moins grande dans le neurone post-synaptique suivant le nombre de récepteurs post-synaptiques présents.

Plusieurs molécules étant libérées lors de la transmission synaptique et plusieurs types de récepteurs pour le même neurotransmetteur pouvant être présents sur la même membrane postsynaptique, plusieurs effets peuvent avoir lieu simultanément sur une synapse donnée. Notamment, sur une même synapse, on peut avoir simultanément des effets excitateur et inhibiteurs.

C.5 Plasticités et apprentissage

C.5.1 Plasticité synaptique

La plasticité synaptique est la faculté des synapses de pouvoir changer la force de leurs effets sur les neurones qu'elles relient. Puisqu'il modifie la structure des circuits cérébraux, il s'agit là d'un des mécanismes clés de l'apprentissage et de la mémoire. Il existe des formes de plasticité à court terme comme à long terme.

On peut classer les différents types de plasticités synaptiques en ces grandes catégories :

- à court terme :
 - l'habituation : la réponse post-synaptique s'atténue au fil des stimuli constants et répétés : ce mécanisme permet d'ignorer un stimulus insignifiant ;
 - la sensibilisation : un stimulus inhabituel déclenche un passage en état d'hypervigilance qui augmente la réaction aux autres stimuli, même neutres ;
- à long terme :

- la potentialisation à long terme : comme dans le cas de la règle de Hebb, le déclenchement d'un neurone par un autre est systématisé ;
- la dépression à long terme : au contraire, le déclenchement est rendu moins probable.

Le potentiel d'action déclenché ou non dans le neurone post-synaptique dépend de l'amplitude de la dépolarisation générée par la synapse. Pour augmenter cette dépolarisation, plusieurs méthodes sont possibles :

- augmentation ou diminution de la quantité de neurotransmetteurs lâchés par les vésicules pré-synaptiques
- augmentation ou diminution du nombre de vésicules synaptiques qui relâchent leurs neurotransmetteurs
- augmentation ou diminution du nombre de vésicules synaptiques dans le neurone pré-synaptique
- augmentation ou diminution du nombre de récepteurs dans le neurone post-synaptique
- augmentation ou diminution de l'importance de la réponse à la réception d'une quantité donnée de neurotransmetteurs
- présence de traces d'ions en plus grande quantité qu'habituellement suite à un stimulus récent
- la dépression à court terme (habituation) peut être provoquée par le temps de recyclage des neurotransmetteurs après épuisement du stock présent dans le neurone pré-synaptique
- il existe également des canaux à ions ne se déclenchant qu'en cas de forte excitation et ayant la propriété de modifier durablement le poids synaptique.

La modification du neurone pré-synaptique requiert une transmission de message dans le sens inverse de la synapse, appelée signal rétrograde. Ce rôle peut être joué entre autres par le monoxyde d'azote, mais les potentiels d'action rétrogrades dont nous avons parlé dans la partie précédente jouent également un rôle.

En plus de cette forme de plasticité qu'on peut qualifier d'homosynaptique (le comportement des neurones reliés par une synapse n'affecte que cette même synapse), existent d'autres formes de plasticité hétérosynaptique (l'activité d'un neurone peut affecter une synapse reliant deux autres neurones). C'est notamment le cas quand un neurone relâche des neuromodulateurs plutôt que des neurotransmetteurs (on l'appelle alors neurone modulateur). Les neuromodulateurs sont des molécules qui vont de diffuser dans de vastes zones du cerveau, et non rester confinées dans une synapse comme les neurotransmetteurs, et qui vont modifier la réponse des synapses aux neurotransmetteurs.

Le premier modèle de plasticité synaptique a été proposé par Donald Hebb en 1949 [Hebb, 1949]. Il s'agit de la règle de Hebb, déduction directe d'observations biologiques, qui modélise le phénomène de potentialisation à long terme. Selon cette règle, quand une cellule A excite par son axone une cellule B et que, de manière répétée et persistante, elle participe à la genèse d'une impulsion dans B, un processus de croissance ou un changement métabolique a lieu dans l'une ou dans les deux cellules, de telle sorte que l'efficacité de A à déclencher une impulsion dans B est,

parmi les autres cellules qui ont cet effet, accrue. On la résume souvent par la phrase "Cells that fire together, wire together". Cependant, cette règle n'est pas un modèle suffisant, puisqu'elle ne permet pas aux connexions de s'affaiblir et ne limite pas non plus leur force. Un modèle de la dépression à long terme est également nécessaire pour éviter une saturation des poids synaptiques qui leur ferait perdre toute sélectivité, et pour éviter une boucle de retour positif dans laquelle un signal s'auto-alimenterait.

Bon nombre d'autres modèles de plasticité synaptique ont en commun de proposer comme variable d'entrée la fréquence de déclenchement du potentiel d'action. Parmi eux, on retrouve le modèle BCM [Bienenstock et al., 1982] (nommé d'après ses concepteurs, Elie Bienenstock, Leon Cooper, et Paul Munro), qui examine la moyenne temporelle de ce taux d'activité et la compare à un seuil. Une réduction d'activité post-synaptique diminue le seuil de potentialisation et augmente celui de dépression à long terme. L'inverse se produit si l'activité post-synaptique augmente.

D'autres modèles de la potentialisation et dépression à long terme choisissent comme variable d'entrée non pas le taux d'activité, mais le moment de déclenchement du potentiel d'action dans les neurones pré- et post-synaptique. C'est ce qu'on nomme la plasticité fonction du temps d'occurrence des impulsions (en anglais Spike-timing-dependent plasticity). En effet, si le potentiel d'action pré-synaptique tend à arriver régulièrement juste avant le déclenchement d'un potentiel d'action post-synaptique, et contribue donc à le causer, alors la connexion synaptique est renforcée. Inversement, s'il arrive juste après, la connexion sera affaiblie. L'existence de cette forme de plasticité a notamment été montrée dans [Markram and Sakmann, 1995, Markram et al., 1997].

Il existe des modèles allant plus loin dans le détail biologique, que nous n'allons pas mentionner ici.

C.5.2 Plasticité neuronale

Les neurones aussi ont leur forme de plasticité, qu'on appelle la plasticité neuronale.

Un neurone peut ainsi modifier son excitabilité intrinsèque, qu'on peut mesurer par son taux de déclenchement de potentiel d'action en réponse à un stimulus. Lorsque l'excitabilité augmente, le neurone répond plus facilement à l'arrivée d'un potentiel post-synaptique et à l'inverse, si l'excitabilité diminue, le neurone y est moins sensible. C'est ce qu'on nomme la plasticité intrinsèque. Pour ce faire, un neurone peut par exemple renforcer ses canaux à ions inhibiteurs pour devenir moins excitable, et ses canaux à ions excitateurs pour l'être davantage. Des modifications temporaires de cette excitabilité peuvent également apparaître suite à une haute fréquence de stimulus, qui ne laissent pas au neurone le temps de pleinement se repolariser, mais si ces stimulus se prolongent, le neurone peut au contraire se désactiver et même mourir [Franklin JL, 1992].

La plasticité homéostatique, au contraire, réfère à la propriété qu'a un neurone de conserver un taux d'excitabilité constant à travers des modifications du réseau qui l'entoure. C'est un phénomène de normalisation qui évite d'avoir des neurones trop

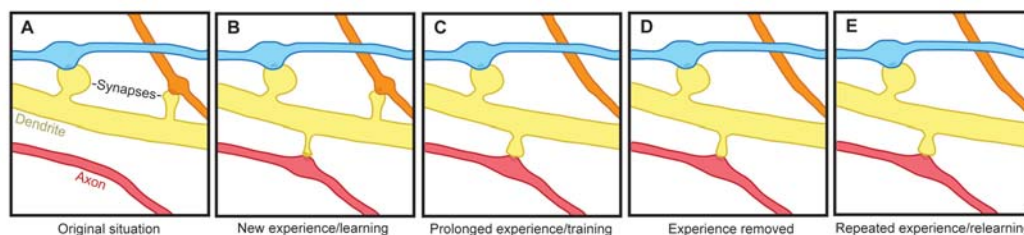


FIGURE C.14 – Plasticité cérébrale

potentialisés ou trop déprimés et permet donc l'équilibre du réseau. Pour ce faire, le neurone peut modifier son excitabilité intrinsèque, comme expliqué précédemment, dans le sens désiré. Le neurone peut également effectuer une mise à l'échelle synaptique ("synaptic scaling"), en modifiant toutes ses synapses entrantes dans le sens désiré, avec les phénomènes de plasticité synaptique cités plus haut [Turrigiano et al., 1998].

Un dernier phénomène est la redistribution du poids des synapses, qui conserve la somme des poids des synapses à travers toute la cellule, mais met ces synapses en compétition entre elles, ne préservant que les plus importantes.

D'un point de vue spéculatif sur les bases cellulaires de la mémoire et de l'apprentissage, la différence entre cette forme de plasticité et la plasticité synaptique est que si cette dernière permettrait théoriquement de retenir une grande quantité d'information (vu le nombre de synapses pouvant être potentiées ou déprimées), seule la modification des propriétés non synaptiques de la membrane permettrait de modifier le codage en fréquence et les caractéristiques temporelles de ce codage.

C.5.3 Plasticité cérébrale

En plus des niveaux précédents de plasticité, les réseaux de neurones et de synapses les reliant peut lui-même évoluer au fil du temps. C'est ce qu'on appelle la plasticité cérébrale.

Le mécanisme principal de la plasticité cérébrale est l'élagage de synapses. Lorsque les synapses sont peu utilisées, elles vont être éliminées, notamment par rétractation ou dégradation de l'axone, ou par disparition et apparition d'épines dendritiques [Maletic-Savatic et al., 1999]. En revanche, une synapse très utilisée sera au contraire renforcée. Ce mécanisme est illustré dans la figure C.14. Ainsi, la configuration du réseau après élagage est plus efficace, n'ayant enlevé que le superflu.

En outre, certains neurones meurent par apoptose, et certains neurones peuvent apparaître, même si la neurogenèse est assez limitée à l'âge adulte.

Ces modifications ont pour conséquence de renforcer les réseaux de neurones dont les décharges se produisent souvent en même temps et de les séparer des autres sous-réseaux de neurones. C'est ce phénomène qui est à l'origine de la séparation en aires corticales fonctionnelles, mentionnées plus haut. Mais la preuve que ces aires sont générées par des mécanismes de plasticité cérébrale est que, chez les handicapés de naissance, la répartition des aires est souvent différente. Ainsi, chez un aveugle de

naissance, les aires habituellement dédiées à la vision seront récupérées par les autres sens, qui pourront ainsi être un peu plus développés. Chez les adultes également, suite à une blessure, des réorganisations structurelles et fonctionnelles peuvent se produire, de façon à récupérer la fonction perdue.

C.6 Structure et assemblage des neurones dans le néocortex humain

L'étude histologique du cortex humain a commencé dès le XIXe siècle, avec notamment Santiago Ramon y Cajal et Camillo Golgi. Ce champ d'études a permis de découvrir que les neurones sont disposés dans le cortex selon la structure d'une grille en deux dimensions comprenant couches et colonnes.

Les neurones sont disposés en six couches horizontales, qui contiennent des types de neurones différents et n'ont pas les mêmes types de liens entre neurones. Chaque couche a une épaisseur semblable d'environ 0,5 mm. Néanmoins, certaines zones du cortex présentent de légères différences, comme par exemple le cortex moteur, qui n'a pas de couche IV. Il est à noter que cette organisation du cortex en couches ne se retrouve pas nécessairement à l'identique chez les autres animaux. Ainsi, le cortex du dauphin a 5 couches, et celui des reptiles n'en a que 3. Ces couches ont été détectées en premier par Ramon y Cajal et par Brodmann [Brodmann, 1909].

Dans la figure C.15, les couches II et III ainsi que les couches IV et V ont été regroupées.

Les couches sont numérotées depuis la surface du cortex vers sa profondeur. On distingue dans l'ordre :

1. la couche moléculaire, qui contient majoritairement des axones et des dendrites. Les axones sont orientés parallèlement à cette surface du cortex, et les dendrite perpendiculairement. Les quelques neurones qui s'y trouvent sont essentiellement des neurones de Cajal-Retzius et des neurones étoilés.
2. la couche granulaire externe, qui contient des neurones granulaires. Elle reçoit les connexions venant des autres zones du cortex.
3. la couche pyramidale externe, qui contient des neurones pyramidaux. Elle émet des connexions vers les autres zones du cortex.
4. la couche granulaire interne, qui contient des neurones étoilés et pyramidaux. C'est là qu'aboutissent les connexions en provenance de l'extérieur du cortex (par exemple du thalamus) ainsi que de l'autre hémisphère cérébral.
5. la couche pyramidale interne, qui contient des neurones pyramidaux ainsi que des neurones pyramidaux géants. C'est de là que partent les connexions qui sortent du cortex, comme par exemple les connexions motrices.
6. la couche polymorphe, qui contient des neurones atypiques. C'est de là que partent les connexions axonales en direction du thalamus.

La couche 4 est également appelée couche granulaire interne, par opposition à la couche 2 qui est la couche granulaire externe. Les couches 1 à 3 sont dites

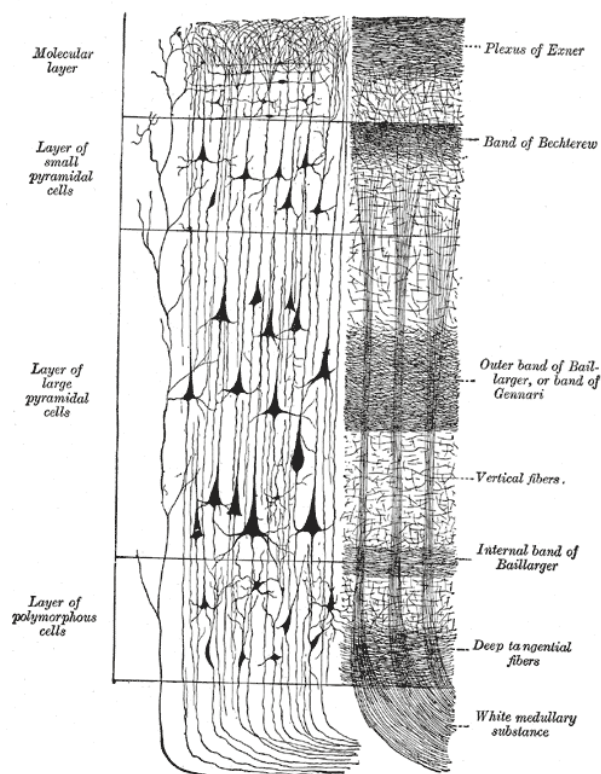


FIGURE C.15 – Couches corticales et leur population de neurones. On observera aussi la forte connectivité verticale qui est à l’origine du concept de colonnes.

supragranulaires, tandis que les couches 5 et 6 sont infragranulaires. Les couches 3 et 5 sont également appelées respectivement couche pyramidale externe et interne.

Une septième couche apparaît brièvement pendant l’embryogénèse, puis disparaît. Une nouvelle étude vient de remarquer chez les autistes un trouble de développement des couches néocorticales sur de nombreuses petites zones dans le cortex temporal et préfrontal, ce qui laisse sous-entendre la crucialité de cette structuration pour un fonctionnement normal du cerveau [Stoner et al., 2014].

Dans les années 1950, Vernon Mountcastle avait observé [Mountcastle, 1957] que la connectivité des neurones n’était pas uniforme. En effet, les neurones sont très connectés avec les neurones des couches supérieures ou inférieures, mais par contre, n’ont pas ou que peu de connexions avec les neurones voisins sur la même couche.

En 1938, Lorente de Nó avait repéré une autre structure anatomique [de No and Fulton, 1938]. Il s’agit des colonnes de neurones, perpendiculaires aux couches et donc à la surface du cortex. Elles concernent surtout les couches 2 à 6, puisque la couche 1 contient peu de neurones. En outre, Vernon Mountcastle a également découvert dans les années 1950 que les neurones sont très fortement connectés entre les différentes couches d’une même colonne, et réagissent au même stimulus au sein d’une colonne. À la structuration anatomique s’ajoute donc une structuration fonctionnelle. Cette forte

C.6. Structure et assemblage des neurones dans le néocortex humain 217

connectivité interne est toute relative, puisque même au sein d'une colonne, environ 90% des synapses proviennent de neurones extérieurs à la colonne.

Historiquement, les colonnes ont d'abord été mises en évidence par des études au sein du cortex visuel, mais sont en fait présentes dans tout le néocortex [Mountcastle, 1997]. Le paléocortex et l'archicortex, structurellement plus simples, ne sont pas stratifiés et ne présentent pas cette structure en colonnes. On peut donc penser que c'est cette structuration qui constitue la spécificité fonctionnelle du néocortex. Le premier, Lorente de Nó avait émis cette hypothèse, reprise par Mountcastle, constatant que tous les types cellulaires connus étaient représentés au sein de ces colonnes, et parlant à leur sujet d'unité élémentaire.

En fait, deux formes de structure en colonne de taille différentes ont été repérées : les mini-colonnes (également appelées micro-colonnes) et les macro-colonnes.

Il y a environ 200 millions de mini-colonnes dans le néocortex [Rakic, 1995]. Les mini-colonnes font environ 30 μm de diamètre et contiennent chacune de 80 à 100 neurones (et deux fois plus dans le cortex visuel), répartis parmi les cinq couches neuronales, et comprenant tous les différents types de neurones vus plus haut. Tous ces neurones réagissent aux mêmes entrées ; par exemple, dans le cortex sensoriel, ces entrées peuvent être le point d'aboutissement d'un neurone sensoriel. Dans d'autres zones, les traitements préalablement faits par le cortex auront permis de déterminer les propriétés concernées. Par exemple, dans l'aire V1 du cortex visuel, on pourra avoir une mini-colonne qui réagit à une ligne orientée à 15° de la verticale, vue par l'œil gauche. Ces structures sont donc aussi bien anatomiques que fonctionnelles [Buxhoeveden and Casanova, 2002].

Le rôle des fonctionnements inhibiteurs est décisif dans la définition des mini-colonnes, puisqu'on retrouve des axones correspondant à des connexions inhibitrices entre les différentes mini-colonnes, les séparant physiquement, et puisque ce fonctionnement est probablement à la base de la réponse aux mêmes entrées, en empêchant les cellules de s'activer lorsqu'une mini-colonne voisine s'active. Les différents neurones d'une même mini-colonne proviennent, d'un point de vue ontogénétique, de la même cellule précurseur neuronal. Si les mini-colonnes sont toutes semblables, elles ne sont toutefois pas identiques au point de vue de l'inventaire exact des cellules présentes, ni des connexions réalisées, puisque ces connexions sont sculptées par les mécanismes de plasticité mentionnés plus haut.

Les macro-colonnes sont estimées au nombre d'un demi-million dans le néocortex, contenant chacune approximativement 60 000 neurones. Elles forment des cylindres de 0,5 mm de diamètre et de 2 mm de profondeur. Elles contiennent de 50 à 100 mini-colonnes, reliées au sein de la macro-colonne par de courtes connexions horizontales. Là encore, leurs champs réceptifs sont assez proches. Cependant, les courtes connexions horizontales ne définissent pas une structure anatomique nette. On peut donc supposer un recouvrement partiel entre macro-colonnes. Selon d'autres, les macro-colonnes correspondent plutôt à une autre structure anatomique, à savoir l'ensemble des mini-colonnes irriguées par un même ensemble de connexions cortico-corticales via les terminaisons arrivant de la couche 1, qui sont mieux définies spatialement. Du fait de leur définition assez floue, l'existence de ces macro-colonnes est remise en cause

par certains, notamment Horton et Adams [Horton and Adams, 2005, Rakic, 2008].

Enfin, les hypercolonnes ne sont pas une structure anatomique, mais un terme employé pour désigner l'ensemble des colonnes correspondant à l'ensemble des valeurs possibles pour un champ réceptif donné, par exemple l'ensemble des orientations possibles pour une ligne dans le cortex visuel [Hubel, 1977]. Ce terme est interchangeable avec celui de module.

Selon la théorie de Vernon Mountcastle [Mountcastle, 1997], ces structures seraient des unités de traitement indépendantes de l'information, chacune recevant et traitant un certain type de stimuli sensoriel provenant d'une partie définie du corps. Ces colonnes sont répétées quasiment à l'identique et représentent l'unité fonctionnelle du néocortex, l'élément de base d'un algorithme massivement parallèle. C'est la théorie de l'organisation en colonne du cortex, dont les détails sont cependant encore débattus.

Cette double structuration en couches et en colonnes nous permet d'identifier un circuit de l'information au sein du néocortex, appuyé sur des observations anatomiques, selon lequel l'information circule verticalement au sein d'une colonne et horizontalement dans la couche 1.

Des faisceaux d'axones en provenance du thalamus aboutissent principalement dans la couche 4, mais également un peu dans les couches 3 et 5 de certains groupes de colonnes, avec moins de terminaisons en dehors de ces groupes. Ces mêmes neurones projettent ensuite leurs axones verticalement au sein de la colonne : les neurones de la couche 4 vont activer les neurones des couches 2 et 3, avec pondération pour la couche 2 par les signaux reçus par d'autres colonnes corticales, via les axones descendant de la couche 1. À leur tour, ces neurones vont activer les neurones des couches 5 et 6. Les cellules de la couche 6 envoient alors des signaux en direction du thalamus, tandis que les neurones moteurs de la couche 5 peuvent transmettre des ordres moteurs. En parallèle, les neurones de la couche 3 transmettent leur activation à d'autres colonnes corticales.

Il est supposé que les connexions vers le thalamus permettent une rétroaction sur les entrées du cortex cérébral.

C.7 La mémoire

Nous avons donc vu les principales structures anatomiques cérébrales, il est maintenant temps d'essayer d'élaborer des hypothèses quant à la façon dont leur agencement permet les principales fonctionnalités de l'esprit humain. En premier lieu, il nous permet pertinent de nous intéresser à la mémoire, qui est évidemment une composante cruciale de l'apprentissage. Comment le cerveau stocke-t-il les informations qu'il accumule, et comment permet-il de les rappeler à la conscience au bon moment ? Plusieurs hypothèses se confrontent pour expliquer la mémoire.

Le courant cognitiviste classique regroupe habituellement sous le terme de mémoire les processus d'encodage, de stockage et de récupération des représentations mentales. Beaucoup de recherches sur la mémoire en psychologie cognitive consistent

à repérer et à décrire ses différents composants. Pour ce faire, les psychologues se basent sur des résultats expérimentaux et sur les symptômes manifestés par des patients cérébrésés.

C.7.1 Emplacement de la mémoire dans le cerveau

Rappelons que les cognitivistes pensent que le cerveau effectue un raisonnement symbolique en manipulant des structures symboliques, à l'aide de règles logiques. Selon cette théorie, le cerveau pourrait donc être composé d'un centre de la mémoire, servant à stocker les symboles, et de différents centres de raisonnement. Cette structure modulaire pourrait ressembler à l'architecture des ordinateurs actuels, dite de Von Neumann, dans laquelle on sépare la mémoire à long terme (le disque dur), la mémoire à court terme (la RAM), et le centre de traitement (le processeur). Le découpage du cerveau en aires fonctionnelles a pu sembler leur donner raison dans un premier temps.

Karl Lashley s'intéresse dès 1910 à l'effet des lésions dans le cerveau. Il étudie les capacités des rats à résoudre certaines tâches ou problèmes après leur avoir prélevé des morceaux de cerveau. Il découvre que le prélèvement d'un morceau du cerveau ne supprime entièrement aucune fonction mais en détériore plusieurs. Il s'aperçoit également d'une part, que quelque soit l'endroit du cerveau qui est prélevé, le résultat en terme de handicap est le même, et d'autre part, que l'ampleur du handicap est linéairement dépendant de la quantité de tissus prélevés. Ces conclusions [Lashley, 1929] sont très surprenantes à l'époque car elles prouvent qu'il n'existe pas de localisation précise pour chaque fonction dans le cerveau. Or, les études visaient plutôt à associer des capacités à des régions cervicales.

Après 20 ans de recherches, Lashley publie en 1950 un ouvrage dans lequel il parvient à expliquer ses résultats de 1929. Dans « In search for the engram » [Lashley, 1950], il définit ce qu'il appelle un engram, ou trace mnésique. Il attribue l'élaboration de la mémoire à des modifications biochimiques des propriétés des synapses au sein des 10 à 12 milliards de neurones du cerveau humain connectés en réseaux de neurones. Il prouve de façon indéniable que les informations dans le cerveau sont distribuées. Il est à l'origine du principe d'équipotentialité, qui affirme que n'importe quelle partie du cortex peut être amenée à jouer n'importe quelle fonction, et qu'une fonction détruite suite à la destruction d'une aire cérébrale peut être reconstruite ailleurs par la suite. En pratique, cette importante plasticité cérébrale n'est possible à grande échelle que durant l'extrême jeunesse d'un individu.

En conséquence, aujourd'hui, le consensus est que l'intégralité du cortex est utilisé pour la mémoire aussi bien que pour le traitement des informations. On pourra trouver les souvenirs visuels dans les aires visuelles, les souvenirs complexes dans les aires associatives, etc. Toutefois, l'hippocampe a été identifié comme jouant un rôle crucial dans la formation de nouveaux souvenirs épisodiques [Scoville and Milner, 1957]. L'hippocampe est d'ailleurs une des seules zones du cerveau où la neurogenèse se maintient à l'âge adulte. Il est également actif en présence de stimuli nouveaux [VanElzakker et al., 2008], et son activation artificielle permet de réveiller d'anciens

souvenirs avec beaucoup de détails [Liu et al., 2012b]. Cependant, il semblerait que son rôle ne soit que transitoire, et que les souvenirs se déplacent ensuite dans le néocortex [Smith and Squire, 2009]. En outre, il ne semble pas affecter la formation de la mémoire procédurale (le savoir-faire).

C.7.2 L'apprentissage

Au XIX^e siècle, les scientifiques pouvaient encore supposer que les nouveaux souvenirs venaient de nouveaux neurones. Cependant, on s'est vite aperçu que les phénomènes de neurogenèse chez l'adulte restaient marginaux (en dehors de l'hippocampe), alors qu'au contraire, de nombreux neurones mouraient et n'étaient donc pas remplacés. Il a donc fallu trouver une autre explication. La première fut suggérée par Santiago Ramón y Cajal, qui, le premier, émit l'hypothèse de la modification de l'efficacité des synapses.

Hebb, un élève de Lashley, raffina ensuite cette idée en expliquant que cette efficacité était augmentée dans le cas de cellules dont les déclenchements étaient rapprochés dans le temps. Quand un axone contribue de façon régulière à l'excitation d'un autre neurone, « une sorte de processus de croissance ou changement métabolique se produit » qui augmente l'efficacité de cette action [Hebb, 1949]. Cette théorie est souvent résumée par l'aphorisme "Cells that fire together, wire together" (les cellules qui déclenchent leurs impulsions au même moment se lient entre elles). Toutefois, cet aphorisme oublie l'aspect de causalité dans le déclenchement, qui est crucial. La plasticité synaptique, dont nous avons parlé plus haut, est donc une forme plus spécifique de la règle de Hebb, où l'accent est mis sur la précédence temporelle du neurone pré-synaptique, et qui détaille également en outre un mécanisme de diminution de la connexion.

Par ailleurs, les mécanismes de comportement évoluent selon la loi de l'effet, exposée pour la première fois par Thorndike, qui dit que les actions menant à un résultat positif sont amenées par l'apprentissage à se produire plus fréquemment dans les situations similaires [Thorndike, 1898]. C'est un mécanisme biologique équivalent à et à l'origine de l'apprentissage par renforcement, dont nous avons parlé dans la partie précédente.

Le principal modèle de l'apprentissage dans le néocortex est donc un mécanisme de triple plasticité : plasticité synaptique, neuronale et cérébrale, dont nous avons parlé plus haut. La potentialisation à long terme des synapses est un mécanisme particulièrement crucial à l'apprentissage, résumé de façon informelle par la loi de Hebb.

Annexe D

Les réseaux de neurones

D.1 Modèles de neurone formel

Afin de pouvoir utiliser certaines méthodes d'apprentissage, il faut remplacer la fonction d'activation de Heaviside dont nous avons parlé précédemment par des fonctions dérivables une ou deux fois si besoin, telles que la sigmoïde ou la tangente hyperbolique. Ces fonctions ayant une sortie continue et non binaire, un seuil doit ensuite être employé afin de décider si le neurone est activé ou non. La figure D.1 présente quelques fonctions d'activation possibles.

Par la suite, d'autres modèles de neurones formels ont été proposés, tels que les neurones à base radiale (RBF : réseaux à fonction radiale de base) [Powell, 1985], les neurones sigma-pi, etc.

Afin de rendre les modèles de réseaux de neurones plus proches de la réalité biologique et de leur attribuer des comportements plus complexes, on a modélisé les réseaux de neurones à impulsions. Leur spécificité est de posséder une importante composante temporelle qu'on ne retrouve pas dans les modèles plus simples. En effet, dans ces modèles, les neurones ne s'activent pas lors de chaque cycle de propagation, mais seulement lorsque le potentiel de leur membrane a atteint un certain seuil, ce qui peut prendre du temps. La communication entre neurones reste la même, à ceci près qu'elle peut atteindre le niveau de détail du fonctionnement d'une synapse.

On préfère donc ici utiliser la méthode de codage consistant à observer le timing précis entre plusieurs impulsions (spikes, en anglais), plutôt que le taux d'impulsions, pour en déduire l'information transmise.

Il existe différents modèles appartenant à la famille des réseaux de neurones à impulsion. Le modèle neuronal le plus ancien est nommé « integrate and fire » (intégrer et déclencher). Il a été proposé dès 1907 par Louis Lapicque [Lapicque, 1907] et est basé sur l'équation simple $I = CdU/dt$ (I = intensité, C = capacité, U = tension), avec un déclenchement d'impulsion à partir d'un certain seuil de tension et une remise à zéro de la membrane après impulsion. On peut rendre ce modèle plus réaliste en lui rajoutant une période réfractaire après chaque impulsion durant laquelle aucune nouvelle impulsion ne peut se produire. Une autre amélioration dans un souci de réalisme est de lui rajouter un mécanisme de fuite de potentiel avec le

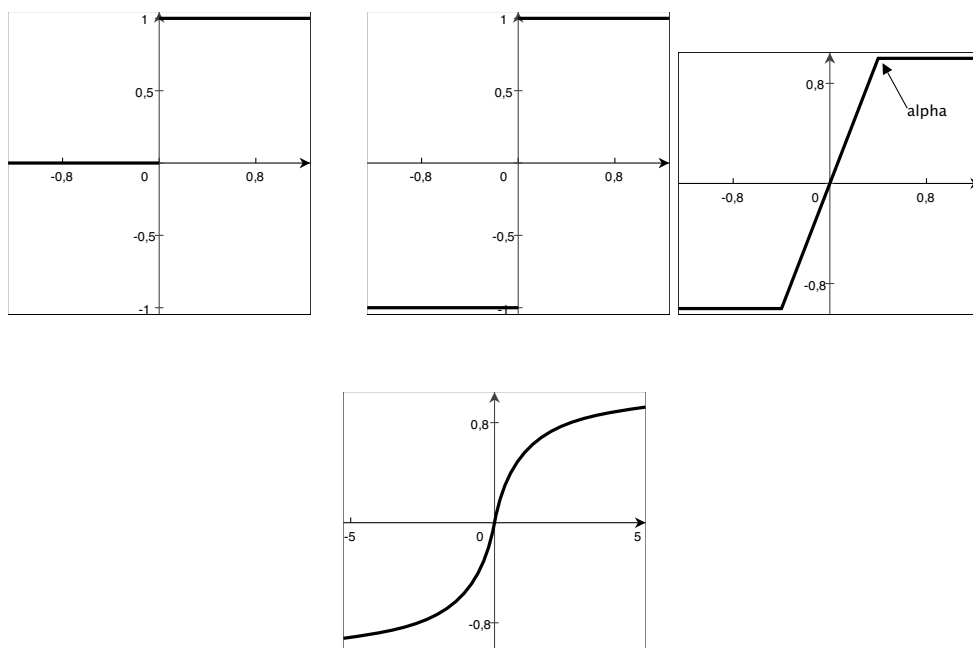


FIGURE D.1 – Diverses fonctions d'activation possibles : fonction de Heaviside, fonction signe, fonction linéaire à seuil, fonction sigmoïde.

temps, pour éviter qu'un boost reçu n'ayant pas déclenché d'impulsion neuronale ne contribue trop à une impulsion beaucoup plus tardive. Ce modèle modifié se nomme « leaky integrate and fire ». Ces deux modèles sont basés sur la modélisation du courant, c'est pourquoi on les qualifie de « current-based » ou CUBA.

Un modèle plus complexe et très employé à ce jour a été proposé par Hodgkin et Huxley en 1952 [Hodgkin and Huxley, 1952]. Il rajoute à notre équation la modélisation du comportement des synapses, où les impulsions sont transmises par des molécules chimiques, les neuro-transmetteurs. Le débit d'ions pouvant passer dans les canaux limite la conductance, qui elle-même influe sur l'intensité. Ce modèle est donc qualifié de « conductance-based », ou COBA.

D.2 Méthodes d'apprentissage

Le plus célèbre modèle de réseau de neurones récurrent est celui de Hopfield, proposé en 1982 [Hopfield, 1982]. Dans ce modèle, le réseau est totalement connecté, hormis qu'il n'existe pas de connexion d'un neurone vers lui-même. De plus, la matrice de connexions est symétrique, c'est-à-dire que le poids de la connexion du neurone A vers le neurone B est égal au poids de la connexion du neurone B vers le neurone A ; cette condition permet de garantir la convergence du réseau. Ce réseau se comporte comme une mémoire adressable de capacité de stockage $N/4\log(N)$, où N est le nombre de neurones dans le réseau. Après apprentissage, un vecteur d'entrée

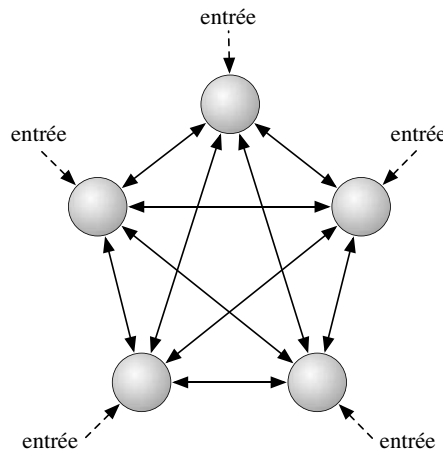


FIGURE D.2 – Le réseau de Hopfield

produira en sortie le résultat mémorisé le plus similaire. Cette mémoire est donc dite auto-associative : on peut évoquer une donnée mémorisée à partir d'une partie de cette donnée. La topographie du réseau de Hopfield est présentée dans la figure D.2.

Hopfield a proposé pour son réseau la mesure d'une quantité qualifiée d'énergie :

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

avec w_{ij} le poids de la connection entre les unités i et j , s_j l'état de l'unité j et θ_i le seuil de l'unité i .

La fonction d'apprentissage choisit fait nécessairement diminuer (non strictement) cette fonction d'énergie, ce qui permet de définir la convergence du réseau par apprentissage comme l'arrivée à un minimum local d'énergie, qui est donc un état stable.

Un modèle semblable au réseau de Hopfield est la machine de Boltzmann. Ce modèle a été proposé par Hinton et Sejnowski en 1985 [Ackley et al., 1985]. Il s'agit aussi d'un réseau de neurones récurrent, intégralement connecté et symétrique, mais sa principale différence est qu'il est stochastique. En effet, sa fonction de transfert est probabiliste. La fonction de transfert généralement utilisée pour cela est la fonction logistique. Cette loi permet au réseau de suivre une distribution de Boltzmann.

$$p_{i=\text{on}} = \frac{1}{1 + \exp(-\frac{\Delta E_i}{T})}$$

avec $\Delta E_i = \sum_j w_{ij} s_j + \theta_i$

ΔE_i représente ici la différence d'énergie globale due au passage de l'unité i de l'état inactif à l'état actif. T , la température, est un paramètre dont la variation peut

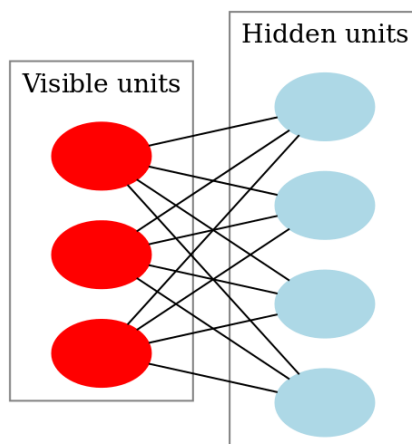


FIGURE D.3 – Une machine de Boltzmann restreinte

permettre de procéder à un recuit simulé. Quant T vaut 0, la machine de Boltzmann est équivalente à un réseau de Hopfield.

Un des problèmes de ce modèle est le temps d'apprentissage exponentiel par rapport à la taille du réseau et la force des connexions. Cela rend l'utilisation des machines de Boltzmann impossible à des tailles non triviales. Pour résoudre ce problème, Smolensky a proposé une variante, la machine de Boltzmann restreinte [Smolensky, 1986]. Dans ce modèle, le réseau est divisé en neurones visibles et neurones cachés. Chacune de ces parties est complètement interconnectée avec l'autre, mais aucune n'a de connexion directe en son sein. Une machine de Boltzmann restreinte est présentée dans la figure D.3.

Un autre exemple de modèle est celui des « echo state network » [Jaeger, 2001], qui comprennent une couche cachée faiblement et aléatoirement connectée (autour de 1% de connectivité). Un exemple en est présenté dans la figure D.4. Les seuls poids pouvant être modifiés sont ceux de la couche de sortie. Ce réseau a une mémoire à court terme dont la capacité de stockage est inférieure à N . Il est utilisé pour apprendre une relation non-linéaire entre entrées et sorties.

Un des problèmes rencontrés avec ce type de mémoire est que le vecteur propre ayant la valeur la plus grande exerce une attraction plus importante. L'utilisation d'une fonction de transfert non linéaire créant de multiples bassins d'attraction peut permettre de contrer ce phénomène. D'autre part, afin d'éviter de tomber dans des minima locaux lors de la phase de convergence, on peut utiliser des algorithmes d'apprentissage tels que le recuit simulé, qui consiste en permettre de choisir des solutions qui ne sont pas meilleures que les précédentes avec une probabilité décroissante au fil du temps.

Stephen Grossberg a développé un autre modèle, l'Adaptive Resonance Theory (ART), à partir de 1976 [Grossberg, 1976, Carpenter and Grossberg, 2003]. Plus qu'un modèle, il s'agit en fait, comme son nom l'indique, d'une théorie; elle soutient

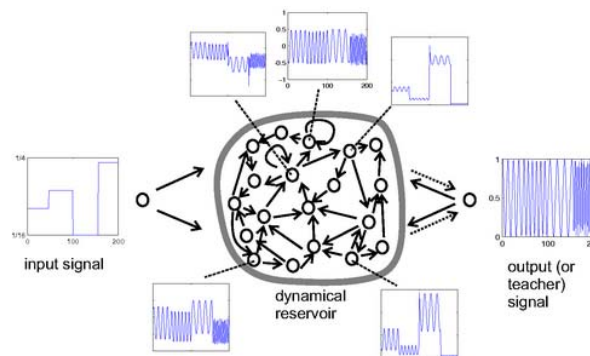


FIGURE D.4 – Un Echo state network

que la reconnaissance et la classification au sein d'un réseau de neurones résulte de l'interaction entre les messages venus du haut, qui correspondent aux attentes et prédictions, et ceux venus du bas, qui transmettent les informations sensorielles. La rencontre de ces deux messages génère un feedback positif amplifiant et prolongeant ainsi l'activation de cet état, qui est donc qualifié de résonant, et qui déclenche l'apprentissage. Selon ses concepteurs, cet état de résonance serait également à l'origine du phénomène de conscience.

En conséquence, un modèle ART possède deux modules, un de reconnaissance et un de comparaison. Un paramètre de vigilance permet de régler le niveau de détail avec lequel sont engrangés les souvenirs. Comme pour les cartes auto-adaptatives, une entrée active le neurone dont l'ensemble des poids est le plus proche d'elle. Cette architecture est montrée à la figure D.5.

Une vigilance faible mène à la création de prototypes abstraits en mémoire et à une grande généralisation. À l'inverse, une forte vigilance permet de stocker de nombreux détails, au détriment de l'abstraction et de la généralisation. Si aucun souvenir stocké ne correspond à une nouvelle entrée dans les limites imposées par la vigilance, un nouveau neurone est affecté à cette entrée.

L'ART prétend ainsi répondre au problème soulevé à l'époque du compromis entre stabilité et plasticité : comment conserver les connaissances acquises tout en maintenant la capacité d'en acquérir de nouvelles. Les deux moyens proposés pour lutter contre cet écueil sont d'avoir un réseau capable de grandir, et d'avoir un réseau composé de deux modules, le premier correspondant à une mémoire à court terme capable de reconnaître un motif qui se présente et le deuxième à une mémoire à long terme capable de généraliser les souvenirs du premier module et en créer de nouveaux.

Les modèles plus récents, LAMINART [Raizada and Grossberg, 2003] et SMART [Grossberg and Versace, 2008], entrent dans des détails biologiques encore plus réalistes en cherchant à reproduire les différences couches du néocortex et les connexions entre elles ainsi qu'avec le thalamus.

Certains modèles s'intéressent également à la reproduction de certains aspects du

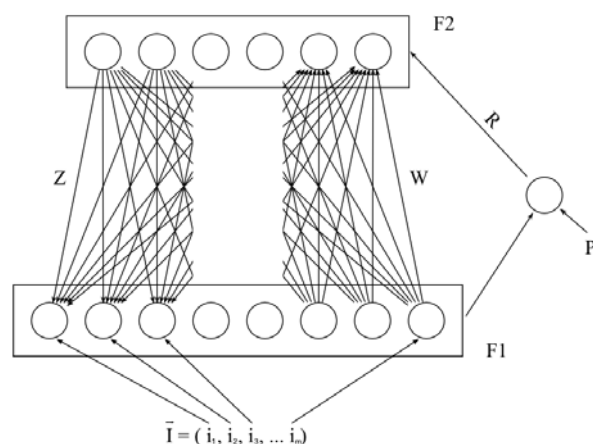


FIGURE D.5 – Les deux modules de l'ART

codage neuronal. C'est le cas du modèle de Olshausen [Olshausen et al., 1996], qui reproduit un codage clairsemé. Les travaux de Földiák [Földiák, 1990] et d'Endres [Endres, 2006] ont également montré la plausibilité d'apprentissage d'un codage clairsemé par un réseau de neurones.

D.3 Méthodes d'apprentissage

Une carte auto-adaptative ne peut néanmoins pas tout représenter à cause de la structure fixe de son réseau ; une liaison entre neurones ne peut être cassée, même pour mieux représenter des données discontinues. Les gaz neuronaux ont été créés dans le but de résoudre ce problème. L'intuition physique derrière ce modèle est d'imiter le comportement d'un gaz remplissant un volume. À partir d'un ensemble de points initial, à chaque nouvel exemple, un lien relie les deux points les plus proches de l'exemple et les points se déplacent par ordre décroissant de proximité avec l'exemple, d'une valeur également décroissante, dans la direction de celui-ci. Un calcul d'obsolescence des liaisons permet d'éliminer celles que le déplacement des points a rendues incorrectes. Suivant les variantes, des unités peuvent être ajoutées au cours de l'apprentissage (modèle de Fritzke [Fritzke et al., 1995], le gaz neuronal croissant) ou non (modèle de Martinetz et Schulten [Martinetz et al., 1991]). Le gaz neuronal croissant permet de bien adapter le nombre de points nécessaires à la bonne représentation des données, et donc de découvrir également des « petits recoins ». La figure D.6¹ présente la différence lors d'un apprentissage de topologies entre un gaz neuronal et un gaz neuronal croissant.

1. Source : [Fritzke et al., 1995]

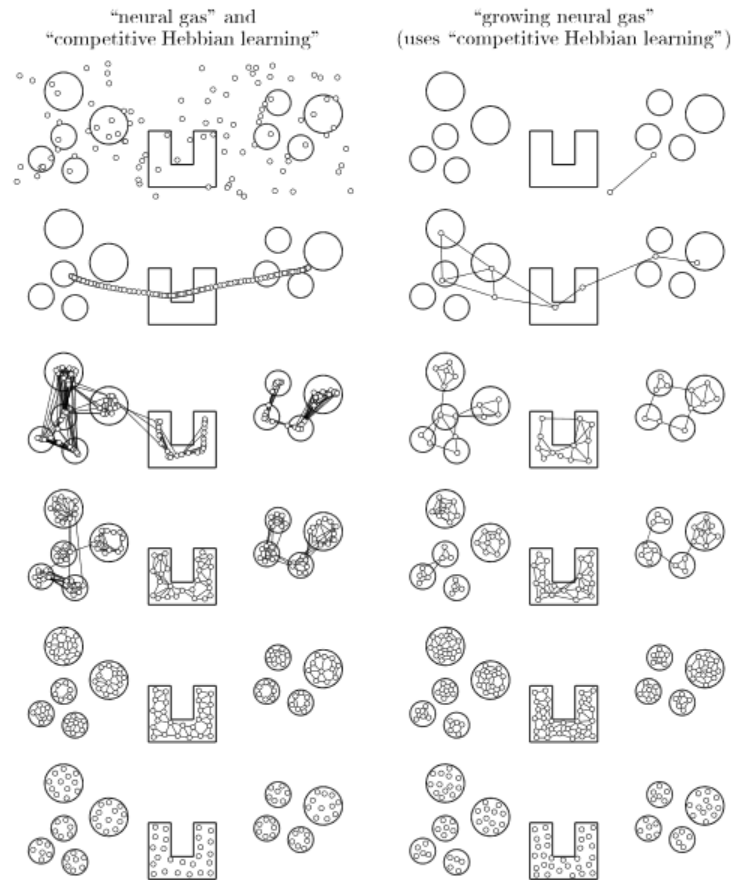


FIGURE D.6 – Apprentissage par un gaz neuronal et un gaz neuronal croissant

Il n'est parfois pas possible d'entraîner un réseau de neurones par apprentissage supervisé, sans qu'il soit pour autant nécessaire de laisser l'apprentissage sans aucune information. Cela peut par exemple arriver si l'on souhaite entraîner un agent intelligent dans un jeu ou un robot à effectuer une certaine tâche. On emploie généralement dans ce cas un apprentissage dit par renforcement, où l'on évalue la performance du réseau sans avoir de couples entrées-sorties prédéfinis.

Parmi les méthodes d'apprentissage par renforcement applicables aux réseaux de neurones, les algorithmes évolutionnaires figurent en bonne place et voient toutes leurs variantes représentées : algorithmes génétiques, stratégies d'évolution, programmation génétique, programmation évolutionnaire...

On distingue en général deux grandes catégories de réseaux évolutionnaires : ceux qui font évoluer uniquement les poids des connexions et ceux modifient également la topologie du réseau. On peut aussi séparer en deux grandes catégories la façon dont les informations servant à construire le réseau de neurones (son génotype) sont encodées : soit le codage du génotype est direct, c'est-à-dire que chaque neurone et

chaque connexion sont représentés explicitement ; soit les données encodées sont les règles ou structures permettant de générer le réseau, dans le cas du codage indirect. Dans ce dernier cas, l'architecture peut être représentée par des grammaires de Kitano, des L-systèmes ou des méthodes de développement cellulaire inspirées de la biologie.

Les modèles les plus connus de réseaux évolutionnaires sont NEAT (NeuroEvolution of Augmenting Topologies) [Stanley and Miikkulainen, 2002] et sa variante HyperNEAT [Stanley et al., 2009] utilisée pour produire des réseaux de plus grande taille, ainsi que EANT (Evolutionary Acquisition of Neural Topologies) [Kassahun and Sommer, 2005]. Ils ont tous deux en commun d'initialiser la population avec des réseaux de petite taille et à la topologie simple. NEAT utilise un algorithme génétique tandis que EANT se base sur la programmation évolutionnaire et les stratégies d'évolution. NEAT permet également une spéciation permettant de comparer des spécimens prometteurs partis dans des voies différentes. HyperNEAT est utilisé avec des réseaux permettant de générer des motifs (CPPN). EANT a développé son propre langage d'encodage de réseaux, intitulé Common Genetic Encoding.

Dans les modèles basés sur des neurones à impulsion, on peut utiliser une règle d'apprentissage dérivée du modèle de plasticité synaptique fonction du temps de déclenchement des impulsions, Spike Timing Dependent Plasticity [Thorpe, 1990]. Cette méthode a pour effet de réduire la latence d'un neurone réagissant à une série d'impulsions : au fil du temps, sa réponse va arriver plus tôt dans la série. Cette règle a été proposée dans un certain nombre de modèles dédiés à son étude, avec généralement une modélisation assez poussée du fonctionnement interne du neurone, mais a aussi été récupérée par d'autres modèles comme SMART.

Annexe E

Test de partitionnement avec HTM-CLA

Comme mentionné dans la partie 4.3.4, nous avons effectué un certain nombre de tests avec notre implémentation de HTM-CLA. Ces tests ont majoritairement pris la forme de tests de partitionnement, puisque ce choix permettait d'observer directement les colonnes actives pour en déduire ce qui se passait dans le réseau.

Nous joignons ici un exemple de fichier correspondant à un tel test.

E.1 Le test

En entrée, nous avons successivement présenté 3 motifs à 10 dimensions. Pour chaque dimension du vecteur, une valeur était choisie pour le motif parmi 3 valeurs possibles, représentées dans les autres motifs. Les motifs étaient ensuite présentés au réseau pour apprentissage de la façon suivante : 1 2 3 2 1 3 1 3 2, où 1 représente 1000 présentations successives du motif 1, etc. Ce grand nombre de présentations successives visait à garantir la convergence de la représentation du motif dans le réseau. Durant la phase de test, les motifs sont ensuite présentés de la façon suivante : 1 2 3, où 1 signifie 10 présentations successives du motif 1, etc.

Nous avons employé une hiérarchie HTM-CLA à 2 niveaux, où le niveau inférieur était de dimension 10 x 3, et le niveau supérieur de dimension 5x1. Nous avons répété l'expérience 100 fois pour avoir une vision globale malgré l'initialisation aléatoire des connexions dans le réseau.

Comme expliqué à la section 4.3.4, nous avons défini différentes sorties qui sont présentée ci-dessous.

E.2 Résultats traités

Nous présentons ici le résultat de divers indicateurs pour nos 100 simulations.

Itération	Temps de convergence (sur 30000)	Nb groupes vus	Nb groupes gardés	Cohérence (1 groupe par motif)	Discrimination (1 motif par groupe)
0	8800	4	4	oui	oui
1	7652	5	5	non	non
2	8800	5	5	oui	oui
3	5875	5	5	non	non
4	8800	3	3	oui	oui
5	5023	4	4	oui	oui
6	6008	5	5	oui	oui
7	7712	5	5	non	non
8	7712	5	5	non	non
9	7712	5	5	non	non
10	8590	5	5	oui	oui
11	8800	5	5	oui	oui
12	5053	5	5	non	non
13	7712	5	5	non	non
14	8800	3	3	oui	oui
15	5875	5	5	non	non
16	8800	4	4	non	oui
17	5023	4	4	oui	oui
18	7712	5	5	non	non
19	7712	5	5	non	non
20	5023	5	5	oui	oui
21	7652	5	5	non	non
22	5023	5	5	oui	oui
23	8800	3	3	oui	oui
24	5023	4	4	oui	oui
25	7712	5	5	non	non
26	8800	4	4	oui	oui
27	5023	4	4	oui	oui
28	8800	4	4	oui	oui
29	8800	5	5	oui	oui
30	8800	3	3	oui	oui
31	5875	5	5	non	non
32	4026	5	5	non	non
33	8800	4	4	oui	oui
34	5023	4	4	oui	oui
35	5053	5	5	non	non
36	5023	4	4	oui	oui
37	8800	4	4	oui	oui
38	8800	4	4	oui	oui
39	8800	4	4	oui	oui

40	6008	5	5	oui	oui
41	7652	5	5	non	non
42	8800	5	5	oui	oui
43	8800	4	4	oui	oui
44	6008	5	5	oui	oui
45	8800	5	5	oui	oui
46	8800	4	4	oui	oui
47	5023	5	5	oui	oui
48	4403	5	5	oui	oui
49	6008	5	5	non	non
50	7652	5	5	non	non
51	7652	5	5	non	non
52	7712	5	5	non	non
53	6008	4	4	oui	oui
54	8800	5	5	oui	oui
55	7652	5	5	non	non
56	7712	5	5	non	non
57	7713	5	5	non	non
58	8800	3	3	oui	oui
59	7712	5	5	non	non
60	4027	5	5	non	non
61	8800	4	4	oui	oui
62	7652	5	5	non	non
63	8800	3	3	oui	oui
64	7712	5	5	non	non
65	5023	5	5	oui	oui
66	7652	5	5	non	non
67	6008	4	4	oui	oui
68	8800	3	3	oui	oui
69	7713	5	5	non	non
70	7652	5	5	oui	oui
71	8800	4	4	oui	oui
72	8800	4	4	oui	oui
73	8800	5	5	oui	oui
74	8800	4	4	oui	oui
75	8800	4	4	oui	oui
76	6008	5	5	oui	oui
77	8800	4	4	oui	oui
78	8800	4	4	oui	oui
79	6008	5	5	oui	oui
80	7652	5	5	oui	oui
81	8800	4	4	oui	oui
82	8800	5	5	oui	oui
83	4027	5	5	non	non

84	7712	5	5	non	non
85	8800	5	5	oui	oui
86	8800	5	5	oui	oui
87	8800	5	5	non	non
88	8800	5	5	oui	oui
89	7652	5	5	non	non
90	4027	5	5	non	non
91	8800	5	5	oui	oui
92	7712	5	5	non	non
93	8800	5	5	oui	oui
94	4026	5	5	non	non
95	8800	4	4	oui	oui
96	8800	5	5	oui	oui
97	8800	5	5	oui	oui
98	8589	4	4	oui	oui
99	8800	4	4	oui	oui

Nombre de groupes	Nombre de fois vus	Nombre de fois gardés
1	0	0
2	0	0
3	7	7
4	29	29
5	64	64

Nous constatons que seules 7 simulations ne reflètent l'apparition que de 3 groupes correspondant aux 3 motifs. En effet, généralement, la représentation oscille au moins un peu avant de se stabiliser. Il est impossible d'obtenir davantage que 5 groupes puisqu'il s'agit du nombre de colonnes présentes dans la région supérieure et que le taux de clairsemage fait qu'au maximum une peut être activée à la fois.

cohérent	63
incohérent	37
discriminant	64
non discriminant	36

En revanche, notre indicateur de cohérence nous montre comment est la situation lors de la phase de test. S'il vaut oui, cela signifie que le test n'a montré que 3 groupes. Cela n'arrive toutefois pas systématiquement. Il reste donc des erreurs de partitionnement. Cette conclusion est confirmée par l'indicateur de discrimination, qui montre que certains groupes sont parfois validés par plusieurs motifs. Notons également que dans tous les cas sauf 1, cohérence et discrimination vont de pair. Cela montre qu'il existe seulement deux grandes sortes de simulation : les bonnes et les mauvaises. Les mauvaises arrivent un peu plus d'un tiers du temps.

10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000 10000
1 100 1 100 10000 1000 1000 100 10 10000 10000 10000 10000 10000 10 1 1 10 1
10000 100 1 10000 1000 100 10000 1000 100 10 100 1 1000 1 1 100 100 100 10 1
10 1000 1000 10000 1000 1000 1 10000 1 10000 10 10000 1000 100 10 1 1000 1000 100
10 100 10 10 10 1000 10 10 100 1 100 10 1000 100 1 10000 10000 1 1 1000 100 100
10000 100 1 100 100 1000 10 10 10000 1000 1 10 100 10000 10 1000 100 1 10 1000 10
10 100 1 100 10000 10000 10000 1000 10000 10000 10 10 100 10 100 1 10 1000 100 10
1 10 10 10 10000 1 100 10000 10000 1000 1000 10000 10000 100 1 10 100 10000 1
10000 100 10000 10 1000 10000 1 10000 10000 1 10000 10 1000 10 10 100 100 1 10000
10000 10 10 1 100 1000 10000 1 1000 1 1 1000 10000 10 10000 1000 10000 10 1000
1000 1 1 100 1 10 10 1000 100 1000 1 1 10 1 100 1 1000 10 1000 100 100 10000 1000
10000 10 10000 10000 10000 10 1 1 1 1000 10000 1000 1 1 100 10 10 100 100 10000
1000 100 10 1000 1 10000 1 1 10000 10 1 10 10000 1 1000 10 10 10 10 100 1 10000
10 10000 10000 100 1000 10 10 1 1 10000 10000 10000 10 10 1 1000 100 100 100
10000 1000 100 100 10000 10000 10000 10 1000 10000 10 10000 1 10 10000 1 1000 1
10 10000 1000 10 1 100 100 100 10000 1000 1 10000 1 10 10 10 1 1000 1 1000
100 100 100 10 1000 1000 1 10 10000 1 100 10000 100 100 10 10 1 1 100 10 10000
1000 1000 100 100 1 1000 10 1 10 10000 1000 100 100 100 100 10 1000 10 1 100 10
1 10 10 1 10 10000 10000 1 10000 10 100 100 1 10000 10000 10 10000 10000 100 1
1000 1 1000 100 100 10 10000 1000 10 1 10 1 10 1 10 100 10 100 10000 100 10 1 100
1 1000 10 1000 1000 100 1 1 10 100 1 1000 1 1000 1 10 1000 10 10000 1 100 10000
1000 10 100 1000 100 100 1000 100 10 1000 1000 10000 10 1000 10000 1 10000 10000
10 10000 1 1000 100 1000 10000 10 1000 1 100 1000 10 10000 100 1 10000 10 1 100
1 1000 100 10000 1 1 1 100 1 1000 1000 10000 10000 100 1000 100 10000 1000
10000 10 100 1000 10000 1000 100 100 1 1000 1 100 100 1 100 10 100 1 10000 10 100
1000 1000 1 10000 1 10000 1000 10000 100 100 100 1 10000 100 100 10000 1000 100
1000 1 100 1 1000 1 1 10000 1 1 10 1000 1000 100 10000 1 1000 100 10 1000 1000 10
1000 1 1000 100 10 100 10000 1 100 1000 10000 100 100 100 100 1000 10 1000 10000
100 10 100 1 1000 100 1 1000 10000 100 10000 10 1 10 1 10 10000 1000 1000 10000
10 100 1 10000 100 10 10000 1000 100 10 100 1 1 10000 10000 100 10000 1 1 1 1000
10 1000 1000 10000 10000 100 1 1000 10000 1000 10 10 100 1 1 100 100 100 1000 1

10000 10000 1000 1 10 100 1000 1 10 100 1000 10 1000 100 100 10 10 1 100 100 1 1 1
10000 10000 100 1000 10000 10 1 1000 100 1000 1000 10 1 10000 1 10000 1 10 100 1
1000 1000 10 1000 1 10 100 1 10000 1000 1000 1 10 1000 1000 100 1 10000 100 1000
100 1 10 10 1000 10000 10000 100 10000 100 1000 1 10 100 1000 1 1000 10000 100
1000 1 10000 10 100 1 100 100 10 100 100 100 10 1 100 10000 1 10 10 1 1000 10000 1
1 1000 10 10 100 1 10 10000 10 1000 10 100 10000 1000 1000 1000 10 10000 10 10 1
1 1000 1 10 100 10000 1000 10 10 100 1000 1000 100 10000 1000 1000 1000 10 1 1000
10 100 1000 100 1 100 10000 1 1000 10000 10 10000 1000 1 1 1000 100 10 1000 10000
100 100 10 10000 1 1 10000 1 100 100 100 100 1000 10 100 10000 1000 1 1000 1000
1000 1000 100 1 100 10000 1000 100 10000 1 1 10000 1 10 10 10 1000 10 10000 1 1
100 1 1 10000 10000 10000 1000 1000 100 10 10 10000 1 1000 1 1000 1 100 1 100 10
10 1 1000 1 10000 10000 1000 100 1 10000 10 10000 1 1 100 10000 10 10000 10000
100 100 10 1 1 10 100 100 1 10000 10 100 100 1 1000 10000 10 10000 1000 10 1 1000
1 10 1000 1 10000 1 10 1000 10 10 100 10 10000 10000 1 1 10 100 100 1000 1000 1
1 10000 10 1000 10000 1 100 10 10 1 1 10000 10000 1 10000 1000 100 10 10 10000 10
100 1 10000 100 10 1 10000 100 10 1 100 10000 1000 10 10 1000 10000 1 100 1000
1000 100 1000 1 10000 100 10 100 1 10000 10 100 100 1000 1000 1000 1 10000 1 10
1000 1000 1 1 100 1 100 100 10 1 10 1 10 10 10000 10 10 1 10000 10000 1000 100
1000 10 10000 1 10000 1 10000 10000 1000 10 10000 10 1 10000 1000 100 10 10
10 100 1 1 10000 10000 10 10000 1 1000 1 10 100 10 100 1000 10000 10 100 1
1000 1 1000 10 10 10 10 10 1000 1000 1000 10 1000 1000 1 100 10 10000 1000 100
1 10000 10 1000 1 1000 10000 10 100 10 1000 1 10 10 100 100 10 1000 1 1 10 10000
1000 1 10000 10 1 10 1000 1000 100 10 100 100 1 100 10000 10 10 10000 10000
1 10000 100 10000 10000 10 10 1000 10000 1 10000 1 10 1 1 1 10000 1 10000 1 10000
10 1000 100 1 10000 10000 1 10 10 1 100 10000 100 100 10000 100 10000 100 10 10 1
1 10000 10000 1 10000 10000 100 10000 1000 100 1 100 1 100 1 1 10000 100 100 100
1 100 100 10000 10 1000 10 10 100 100 1 100 1 100 1 10 10 10 10000 100 100 10000
100 1000 10000 1000 100 100 1000 10 1000 1000 10000 1 10000 10000 10 100 1 10 1
10 1 1 10000 1000 10 1000 1000 10000 10000 10 10 10 1000 100 100 100 1000 10
10000 10 1 1 1000 1 1 1000 100 10 1000 10 10 10 1000 100 1000 1000 100 100 1000
10000 100 1 1000 100 100 1 1000 10000 10 10000 1 100 1 100 100 100 10

100 1 1000 1000 10 10 10 10000 10 100 10 1000 1000 10 1000 100 10 10 10 10 10000
10 1000 1 100 10000 10000 100 100 1 10000 10 10000 10 10000 1000 10000 10000 100
1 100 10 10 100 1 100 1 10 100 100 10000 1 10 100 1 1000 1 10 10000 1 10000 10 100
100 1 100 1 1 10000 10000 10000 1000 1 10 100 10000 1 100 10000 100 10000 1000 10
10 1000 1 10000 1 1000 1000 1000 1000 10000 10000 1000 10 10 10 1000 1000 10000
100 10000 1000 100 10 1000 10000 10 10 1000 100 1000 100 10 10 10 1000 10000
100 1 10000 1000 10000 1000 10000 100 10000 10000 1000 1 1000 1 10 100 100 1000
10000 100 1 100 100 10000 10 10 1000 10 1000 1000 10 10000 10 1 100 1 10000 10000
100 1 10000 100 1000 1 1 100 100 100 1 10000 10000 10000 1 1000 10 10 1000 1 100
10000 100 10 10000 1 1 100 100 10 1000 10 1000 1 1000 1 1 1 1 10000 1 100 100
10000 100 10 1000 100 100 100 10 10000 10 10 100 1 10000 10 10 1000 10000 1 1000
1000 1 10 10000 10 10000 1000 100 10 1000 100 10000 1 100 10000 1 1 10000 100 100
1000 10000 10000 10 100 1 1000 10 10000 10000 10 1 1000 1000 10 100 10000 1000
100 100 10 1 10 1000 1 10000 100 1 1 1000 10 1 100 1 100 1 100 10000 1000 100 1
1000 1000 1000 10 10 100 10 100 100 10000 10 1000 100 10 100 100 100 1 1 1000 1
10 1 100 10 10 1 100 1000 1 10 1000 1000 100 1000 1 1000 10 10 1 10 10 10000 10 10
1000 100 1000 100 100 100 10000 10000 1 10 1 100 100 1000 1000 100 1000 100 10
100 100 10000 100 1 10000 10000 10 100 10000 100 1000 100 1 10 10000 10000 10000
1000 1 100 100 1 10 100 1 100 10000 1000 10 10000 1000 1000 1 1 10000 1 1 100 100
100 100 1 10000 1000 100 1000 10000 1000 10 1 100 1000 1 10000 10000 1 100 10 10
100 10 10 10000 100 1000 10000 10000 100 100 1000 100 1000 10000 10000 1000 1
100 1 100 1000 10 10 10 100 10000 10 100 10000 10000 10 10 1000 100 10000 100 100
10000 1 100 100 1 10 10 10 10000 1 100 1000 10 100 1000 1 10 1 10000 10000 1000 1
1000 10 1 1000 100 1 10 1000 10 1 10000 100 1000 1000 1000 100 1 10 10 1
100 1 1000 1000 1 10 10 1 10 1000 1000 10000 10 1 10000 10 1 10 1 1 1 100 10 10
100 1 10000 10000 10 1000 10000 1 100 10000 1 1 1000 10000 100 10000 1000 1 10
10000 1 1000 1 100 10 1 100 10000 100 100 1 1000 1 1 100 10 10000 10000 10 10000
1000 1000 10 1000 1 1 1000 10 10 100 1 10 1000 10000 10000 10 100 10000 100 100 1
100 10000 1 10000 1000 1 10 1000 1 100 100 10000 100 1000 100 100 100 10000 100
10 100 10000 10000 1 100 1000 10000 10 100 10 1 1000 1 1000 10 100 10000 1000 1
100 100 10000 10 1000 1000 1 1 10000 1000 1000 1000 10000 10 10000 100 100 1000
1000 10 100 1000 1000 1 1000 10 1 1 10 10 10000 1 100 10 1000 10 100 100 10000

1000 10 10 1000 1000 100 100 10000 1000 100 10000 1000 10000 1 10 1 100 1 100
1000 10000 10000 10000 10 10000 100 10000 10000 1 10 10 10000 1000 10 1000 1 100
10000 1000 1 1000 1 1000 1000 1 1000 1000 10 1 1 10000 100 10 10 100 1 10000 10
10000 10 100 10000 100 1 1000 100 1 100 1 10 1000 10 1000 1 10000 10 1000 100 10
100 10000 10 10000 100 1 10000 1000 1 100 10 100 10000 1000 10 100 10 100 100
1000 1 1 1 100 100 1000 100 1 1000 10 10 10 10000 10000 1000 100 100 1 1000 10000
10 1000 100 1 100 10 100 10 100 1 10 100 100 10000 10000 1 1000 1000 10 10
100 100 1 10 100 1000 10000 1 10000 100 1000 1000 10 100 10 100 1 1000 10 10000
10 100 1000 10 1000 100 10 1 1 10 10 1 10 10 10000 10 1000 1000 1000 10000 10 1
100 10 10000 1 10000 1000 1000 100 10000 10000 100 1 1 1000 100 1000 1000 100
1000 1000 10 1000 10000 10000 1 1000 10 10 1 1000 100 10000 10 1 1000 10000 1000
1 10000 10 10000 1000 1 1 10 10000 10 1 10000 1000 1000 1000 1000 100 1 10 10 1
100 1 1000 1000 1 10 10 1 10 1000 1000 10000 10 1 10000 10 1 10 1 1 1 100 10 10
100 1 10000 10000 10 1000 10000 1 100 10000 1 1 1000 10000 100 10000 1000 1 10
10000 1 1000 1 100 10 1 100 10000 100 100 1 1000 1 1 100 10 10000 10000 10 10000
1000 1000 10 1000 1 1 1000 10 10 100 1 10 1000 10000 10000 10 100 10000 100 100 1
100 10000 1 10000 1000 1 10 1000 1 100 100 10000 100 1000 100 100 100 10000 100
10 100 10000 10000 1 100 1000 10000 10 100 10 1 1000 1 1000 10 100 10000 1000 1
100 100 10000 10 1000 1000 1 1 10000 1000 1000 1000 10000 10 10000 100 100 1000
1000 10 100 1000 1000 1 1000 10 1 1 10 10 10000 1 100 10 1000 10 100 100 10000

```

10000 100 10000 1 10000 10000 10000 1000 10000 10000 10 100 100 1 100 10 10 1
10000 10000 10 1000 10 1000 10 10 10 1 1 10000 10 100 1000 100 1000 10 1000 10
10 1000 1000 1000 1000 100 1 1000 1 1 1 1000 1 10000 1000 10000 1 1000 100 100
1000 10000 10000 1000 1 1 100 1 1000 100 10000 1000 1 10000 100 100 1000 100 100
10000 1000 1000 10 1000 1 1000 1000 1000 10 10 10000 10 10000 10000 1000 100 10
10000 10000 100 1 100 10 10000 100 10 100 1 1000 100 10 10000 1000 10 10 1 1 10
1000 10000 10000 10 100 10 100 10000 1 10000 1 1 1000 100 1000 1 1 1000 10 1000
10 10000 1 10000 1 100 100 1000 1 10000 100 10000 10 10000 10000 10 10000 100
10 1000 10 10000 10 1 10 10000 10 1000 10 10000 1 10 100 1 10 10 10000 10000 1 1
100 10000 1000 10000 100 1000 100 100 1000 1 10 10 1000 100 10000 100 100 1 10
10 1000 100 1 10000 1 10 1 1 10 10 1 10000 100 1 1000 100 1000 10 1000 10000 1 1
100 10000 10000 10000 10000 10000 100 1000 1 1 10000 10 1000 10 10000 1000 1000
10000 10 1000 10 10000 10 100 10 1 10000 10000 10000 1000 10000 100 100 10000 10
1000 1 1 10000 10000 1000 1000 10000 10000 100 10 1000 100 10 10000 100 1000
10000 1000 1 10 10 1000 10 100 100 1000 1 10000 100 10 100 1000 1 10 10 1000 100
10 10000 10 1000 1 1000 100 100 1 10000 10 10000 1 10 10000 100 1 10000 1000 1
100 1000 10000 1000 100 10 10 1 100 10000 10 1000 10000 10000 10 100 100 100 1
10 10000 100 100 10 1 100 1 1 10000 10 1000 10000 1 10 100 10000 10000 100 1000
10000 1 100 10 1000 1 10000 10000 10 1 10000 10000 1000 1000 1 100 1000 10000 10
1000 100 1 10 10000 10 1 1000 100 10000 100 10000 10000 1 1 1 10000 10000 10 10 1
1 10000 10 1000 1 1 1000 10 10000 1 1000 100 1000 1 1 1 10 1 10 100 10000 10000 10
10000 10 10 10 100 10 100 10 10 1000 1 10 1 1000 1 100 10000 1000 1 1 100 1000
1000 10 10000 10 10000 10000 100 1000 1 1 10 1000 100 10 1000 10000 100 1 10 100
100 10000 10000 100 1000 1 1000 10 10 10 1 100 100 10000 1000 1000 10000 1 1 100
1000 1 1 100 10 10000 10 10000 100 100 10 10000 100 100 10000 1 1 1000 1000 10000
1000 100 10 10 100 10000 10 10 100 1 10000 1 10000 10000 10 10 10000 10 10000 10
1 1000 10000 1000 100 1000 10 10000 100 100 1000 1000 10 100 100 100 10000 1 1 10
1000 1000 10000 10000 1 10000 10000 1 10000 1000 10 100 10 10000 1 1 10000 10000
10 1000 10000 10 10000 1000 10000 10 100 1000 1000 1000 1000 10 1 10000 1000
1000 1 100 10 10 10 1 10 100 10 10000 100 1 10 10000 10000 10000 10 1000 10 1 100
1 10000 10 1000

```

Test :

```

100 1 1000 10000 100 10 100 10000 10000 1 1 10000 10 10000 1 1 10000 100 10000
100 100 1000 100 1 1 100 10000 10 10000 10

```

Ici, on voit que la colonne activée change même lorsque le motif présenté ne change pas, et qu'aucune tendance à l'émergence d'un groupe donné pour un motif donné n'apparaît. La représentation n'a donc pas été stabilisée et ne donne aucun signe de le devenir dans un futur proche.

Annexe F

Publications

En 2010, nous avons présenté nos idées sur l'utilisation de HTM pour la génération de comportement lors des journées de l'Association française de réalité virtuelle (AFRV). Les actes de cette conférence n'ont pas été publiés, mais nous laissons toutefois disponible le texte de l'article qui nous a valu d'être sélectionnés [Bovo et al., 2010].

En 2013, nous avons réalisé nos expériences de partitionnement de données, qui nous ont permis plusieurs publications : une première [Bovo et al., 2013a] en tant qu'article court associé à un poster pour la conférence Educational Data Mining (EDM 2013) ; et un article plus détaillé, présenté une fois en français [Bovo et al., 2013c] à la conférence EIAH & IA 2013, et une fois en anglais [Bovo et al., 2013b] à la International Conference on E-Learning and E-Technologies in Education (ICEEE 2013).

Nous avons également été sollicités pour décrire notre projet pour un bulletin spécial de l'AFIA sur la thématique EIAH & IA, qui n'a pas encore été publié.

Enfin, un de nos articles a été cité par un confrère dans un article de revue sur les utilisations du partitionnement de données éducatives [Dutt et al., 2014]. Nous avons également vu deux de nos articles cités dans un mémoire de master [Vermeulen et al., 2014] sur l'utilité pour les élèves d'avoir un retour sur leur progrès et de pouvoir se fixer des buts.

Bibliographie

- [Ackley et al., 1985] Ackley, D. H., Hinton, G. E., and Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169. (Cité en page 223.)
- [Adrian and Zotterman, 1926] Adrian, E. D. and Zotterman, Y. (1926). The impulses produced by sensory nerve-endings part ii. the response of a single end-organ. *The Journal of physiology*, 61(2):151–171. (Cité en page 80.)
- [Agrawal et al., 1994] Agrawal, R., Srikant, R., et al. (1994). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499. (Cité en page 64.)
- [Akaike, 1974] Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723. (Cité en page 157.)
- [Amadiou et al., 2008] Amadiou, F., Bastien, C., and Tricot, A. (2008). Les méthodes on-line 1: Analyse des parcours. *Ergonomie des documents électroniques*, pages 251–270. (Cité en page 52.)
- [Ananthanarayanan and Modha, 2007] Ananthanarayanan, R. and Modha, D. S. (2007). Anatomy of a cortical simulator. In *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, SC '07*, pages 3:1–3:12, New York, NY, USA. ACM. (Cité en page 113.)
- [Anderson, 1993] Anderson, J. R. (1993). Rules of the mind. *Lawrence Erlbaum Associates, Hillsdale, New Jersey*. (Cité en page 51.)
- [Anderson et al., 1989] Anderson, J. R., Conrad, F. G., and Corbett, A. T. (1989). Skill acquisition and the lisp tutor. *Cognitive Science*, 13(4):467–505. (Cité en pages 51 et 52.)
- [Ankerst et al., 1999] Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. In *ACM SIGMOD Record*, volume 28, pages 49–60. ACM. (Cité en page 64.)
- [Anozie and Junker, 2006] Anozie, N. and Junker, B. W. (2006). Predicting end-of-year accountability assessment scores from monthly student records in an online tutoring system. In *Educational Data Mining: Papers from the AAAI Workshop*. AAAI Press. (Cité en pages 32 et 55.)
- [Antibi and Luciani, 2003] Antibii, A. and Luciani, S. (2003). *La constante macabre, ou, Comment at-on découragé des générations d'élèves?* Math'Adore. (Cité en page 18.)

- [Arel et al., 2009] Arel, I., Rose, D., and Karnowski, T. (2009). A deep learning architecture comprising homogeneous cortical circuits for scalable spatiotemporal pattern inference. In *NIPS 2009 Workshop on Deep Learning for Speech Recognition and Related Applications*. (Cité en page 110.)
- [Arel et al., 2010] Arel, I., Rose, D. C., and Karnowski, T. P. (2010). Deep machine learning—a new frontier in artificial intelligence research [research frontier]. *Computational Intelligence Magazine, IEEE*, 5(4):13–18. (Cité en page 153.)
- [Ayers et al., 2009] Ayers, E., Nugent, R., and Dean, N. (2009). A comparison of student skill knowledge estimates. *International Working Group on Educational Data Mining*. (Cité en page 53.)
- [Azevedo et al., 2009] Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Filho, W. J., Lent, R., and Herculano-Houzel, S. (2009). Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology*, 513(5):532–541. (Cité en page 198.)
- [Baker et al., 2004a] Baker, R. S., Corbett, A. T., and Koedinger, K. R. (2004a). Detecting student misuse of intelligent tutoring systems. In *Intelligent tutoring systems*, pages 531–540. Springer. (Cité en page 53.)
- [Baker et al., 2004b] Baker, R. S., Corbett, A. T., Koedinger, K. R., and Wagner, A. Z. (2004b). Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390. ACM. (Cité en page 20.)
- [Baker and Yacef, 2009] Baker, R. S. and Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM—Journal of Educational Data Mining*, 1(1):3–17. (Cité en page 48.)
- [Baradwaj and Pal, 2012] Baradwaj, B. K. and Pal, S. (2012). Mining educational data to analyze students’ performance. *arXiv preprint arXiv:1201.3417*. (Cité en page 54.)
- [Barker et al., 2004] Barker, K., Trafalis, T., and Rhoads, T. R. (2004). Learning from student data. In *Systems and Information Engineering Design Symposium, 2004. Proceedings of the 2004 IEEE*, pages 79–86. IEEE. (Cité en page 54.)
- [Barolli et al., 2006] Barolli, L., Koyama, A., Durrezi, A., and De Marco, G. (2006). A web-based e-learning system for increasing study efficiency by stimulating learner’s motivation. *Information Systems Frontiers*, 8(4):297–306. (Cité en page 20.)
- [Barrette, 2004] Barrette, C. (2004). Vers une métasynthèse des impacts des TIC sur l’apprentissage et l’enseignement dans les établissements du réseau collégial québécois. De la recension des écrits à l’analyse conceptuelle. *Bulletin collégial des technologies de l’information et des communications*, (55):8–15. (Cité en page 16.)
- [Barros and Verdejo, 2000] Barros, B. and Verdejo, M. F. (2000). Analysing student interaction processes in order to improve collaboration. the degree approach. *International Journal of Artificial Intelligence in Education*, 11(3):221–241. (Cité en page 35.)

- [Bastos et al., 2012] Bastos, A. M., Usrey, W. M., Adams, R. A., Mangun, G. R., Fries, P., and Friston, K. J. (2012). Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711. (Cité en page 105.)
- [Beal et al., 2006] Beal, C. R., Qu, L., and Lee, H. (2006). Classifying learner engagement through integration of multiple data sources. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 151. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999. (Cité en page 20.)
- [Beaudoin, 2002] Beaudoin, M. (2002). Learning or lurking? Tracking the "invisible" online student. *The Internet and Higher Education*, (2):147–155. (Cité en page 53.)
- [Beck and Woolf, 2000] Beck, J. E. and Woolf, B. P. (2000). High-level student modeling with machine learning. In *Intelligent tutoring systems*, pages 584–593. Springer. (Cité en pages 47 et 50.)
- [Behnke and Rojas, 1998] Behnke, S. and Rojas, R. (1998). Neural abstraction pyramid: A hierarchical image understanding architecture. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 2, pages 820–825. IEEE. (Cité en pages 94 et 97.)
- [Bellman, 1957] Bellman, R. (1957). E. 1957. dynamic programming. *Princeton University Press. Bellman Dynamic programming 1957*. (Cité en page 151.)
- [Bengio, 2009] Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127. (Cité en page 94.)
- [Berendt and Brenstein, 2001] Berendt, B. and Brenstein, E. (2001). Visualizing individual differences in web navigation: Stratdyn, a tool for analyzing navigation patterns. *Behavior Research Methods, Instruments, & Computers*, 33(2):243–257. (Cité en page 52.)
- [Bienenstock et al., 1982] Bienenstock, E. L., Cooper, L. N., and Munro, P. W. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *The Journal of Neuroscience*, 2(1):32–48. (Cité en page 213.)
- [Boechler and Dawson, 2002] Boechler, P. M. and Dawson, M. R. (2002). Effects of navigation tool information on hypertext navigation behavior: A configural analysis of page-transition data. *Journal of Educational Multimedia and Hypermedia*, 11(2):95–115. (Cité en page 52.)
- [Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM. (Cité en page 62.)
- [Bovo et al., 2010] Bovo, A., Sanchez, S., Héguy, O., and Duthen, Y. (2010). Création de comportements à partir de motifs hiérarchiques dans un contexte temporel. Journées de l'AFRV 2010 (actes non publiés). (Cité en pages 135 et 257.)
- [Bovo et al., 2013a] Bovo, A., Sanchez, S., Héguy, O., and Duthen, Y. (2013a). Analysis of students clustering results based on moodle log data. In *Proceedings of the*

- 6th International Conference on Educational Data Mining (EDM 2013)*. (Cit  en pages 164 et 257.)
- [Bovo et al., 2013b] Bovo, A., Sanchez, S., H guy, O., and Duthen, Y. (2013b). Clustering moodle data as a tool for profiling students. In *Proceedings of the Second International Conference on e-Learning and e-Technologies in Education (ICEEE), 2013*, pages 121–126. IEEE. (Cit  en pages 38, 164 et 257.)
- [Bovo et al., 2013c] Bovo, A., Sanchez, S., H guy, O., and Duthen, Y. (2013c). L’apprentissage automatique comme base du suivi d’ l ves et de l’am lioration de formations. In *Actes de la Journ e EIAH&IA 2013*. (Cit  en pages 38, 164 et 257.)
- [Box et al., 1976] Box, G. E., Jenkins, G., and Reinsel, G. (1976). Time series analysis, forecasting and control. *HOLDEN-DAY SERIES IN TIME SERIES ANALYSIS AND DIGITAL PROCESSING*. (Cit  en page 58.)
- [Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. (Cit  en page 62.)
- [Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press. (Cit  en page 62.)
- [Breunig et al., 2000] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM. (Cit  en page 64.)
- [Brodmann, 1909] Brodmann, K. (1909). *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues*. Barth. (Cit  en page 215.)
- [Broisin et al., 2007] Broisin, J., Vidal, P., et al. (2007). Une approche conduite par les mod les pour le tra age des activit s des utilisateurs dans des eiah h t rog nes. *Revue des Sciences et Technologies de l’Information et de la Communication pour l’Education et la Formation (STICEF)*, 14. (Cit  en page 30.)
- [Bryson Jr, 1969] Bryson Jr, A. E. (1969). Ho-yu-chi. *Applied Optimal Control. Optimization, Estimation and Control*. (Cit  en page 85.)
- [Burnham and Anderson, 2002] Burnham, K. P. and Anderson, D. R. (2002). *Model selection and multimodel inference: a practical information-theoretic approach*. Springer. (Cit  en page 158.)
- [Buxhoeveden and Casanova, 2002] Buxhoeveden, D. P. and Casanova, M. F. (2002). The minicolumn hypothesis in neuroscience. *Brain*, 125(5):935–951. (Cit  en page 217.)
- [Calvino, 2006] Calvino, B. (2006). *Qu’est-ce qu’un neurone ? Les petites pommes du savoir*. Le Pommier. (Cit  en page 205.)
- [Calvo-Flores et al., 2006] Calvo-Flores, M. D., Galindo, E. G., Jim nez, M. C. P., and Pi eiro, O. P. (2006). Predicting students’ marks from Moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, page 1:586–590. (Cit  en page 54.)

- [Carini et al., 2006] Carini, R. M., Kuh, G. D., and Klein, S. P. (2006). Student engagement and student learning: Testing the linkages*. *Research in Higher Education*, 47(1):1–32. (Cité en page 20.)
- [Carpenter and Grossberg, 2003] Carpenter, G. A. and Grossberg, S. (2003). *The Handbook of Brain Theory and Neural Networks, Second Edition*, chapter Adaptive resonance theory, pages 87–90. MIT Press, Cambridge, MA, arbib edition. (Cité en page 224.)
- [Cen et al., 2006] Cen, H., Koedinger, K., and Junker, B. (2006). Learning factors analysis—a general method for cognitive model evaluation and improvement. In *Intelligent Tutoring Systems*, pages 164–175. Springer. (Cité en page 51.)
- [Chen et al., 1998] Chen, G.-D., Shen, G.-Y., Ou, K.-L., and Liu, B.-J. (1998). Promoting motivation and eliminating disorientation for web based courses by a multi-user game. (Cité en page 20.)
- [Chomsky, 1959] Chomsky, N. (1959). A review of bf skinner’s verbal behavior. *Language*, 35(1):26–58. (Cité en page 70.)
- [Choquet et al., 2007] Choquet, C., Iksal, S., et al. (2007). Modélisation et construction de traces d’utilisation d’une activité d’apprentissage: une approche langage pour la réingénierie d’un eia. *Revue des Sciences et Technologies de l’Information et de la Communication pour l’Education et la Formation (STICEF)*, 14. (Cité en pages 21, 29 et 171.)
- [Church, 1935] Church, A. (1935). An unsolvable problem of elementary number theory. In *Bull. Amer. Math. Soc.*, volume 41, pages 332–333. (Cité en page 70.)
- [Claeskens and Hjort, 2003] Claeskens, G. and Hjort, N. L. (2003). The focused information criterion. *Journal of the American Statistical Association*, 98(464):900–916. (Cité en page 158.)
- [Clancey, 1997] Clancey, W. J. (1997). *Situated Cognition: On human knowledge and Computer Representations*. Cambridge University Press. (Cité en page 75.)
- [Cocea and Weibelzahl, 2011] Cocea, M. and Weibelzahl, S. (2011). Disengagement detection in online learning: validation studies and perspectives. *Learning Technologies, IEEE Transactions on*, 4(2):114–124. (Cité en pages 20 et 55.)
- [Cole and Foster, 2007] Cole, J. and Foster, H. (2007). *Using Moodle: Teaching with the popular open source course management system*. O’Reilly Media, Inc. (Cité en page 21.)
- [Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM. (Cité en page 91.)
- [Corbett and Bhatnagar, 1997] Corbett, A. T. and Bhatnagar, A. (1997). Student modeling in the act programming tutor: Adjusting a procedural learning model with declarative knowledge. *COURSES AND LECTURES-INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES*, pages 243–254. (Cité en page 51.)

- [Cox, 1946] Cox, R. T. (1946). Probability, frequency, and reasonable expectation. (Cité en page 155.)
- [Davier, 2008] Davier, M. (2008). A general diagnostic model applied to language testing data. *British Journal of Mathematical and Statistical Psychology*, 61(2):287–307. (Cité en page 52.)
- [Dayan et al., 1995] Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, 7(5):889–904. (Cité en page 98.)
- [de No and Fulton, 1938] de No, R. L. and Fulton, J. F. (1938). *Architectonics and structure of the cerebral cortex*. Oxford University Press. (Cité en page 216.)
- [de Wit et al., 2010] de Wit, L., Machilsen, B., and Putzeys, T. (2010). Predictive coding and the neural response to predictable stimuli. *The Journal of Neuroscience*, 30(26):8702–8703. (Cité en page 100.)
- [Dekker et al., 2009] Dekker, G. W., Pechenizkiy, M., and Vleeshouwers, J. M. (2009). Predicting students drop out: A case study. *International Working Group on Educational Data Mining*. (Cité en page 54.)
- [Dempster et al., 1977] Dempster, A. P., Laird, N. M., Rubin, D. B., et al. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society*, 39(1):1–38. (Cité en page 64.)
- [Desimone et al., 1984] Desimone, R., Albright, T. D., Gross, C. G., and Bruce, C. (1984). Stimulus-selective properties of inferior temporal neurons in the macaque. *The Journal of Neuroscience*, 4(8):2051–2062. (Cité en page 82.)
- [Desmarais et al., 1995] Desmarais, M. C., Maluf, A., and Liu, J. (1995). User-expertise modeling with empirically derived probabilistic implication networks. *User modeling and user-adapted interaction*, 5(3-4):283–315. (Cité en page 51.)
- [Desmarais and Pu, 2005] Desmarais, M. C. and Pu, X. (2005). A bayesian student model without hidden nodes and its comparison with item response theory. *International Journal of Artificial Intelligence in Education*, 15(4):291–323. (Cité en page 54.)
- [Diagne, 2008] Diagne, F. (2008). Em-agiir: un environnement multi-agent ouvert pour la supervision à partir d’indicateurs réutilisés. *RJCEIAH2008*. (Cité en page 30.)
- [DiCarlo et al., 2012] DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron*, 73(3):415–434. (Cité en pages 97 et 101.)
- [Dimitracopoulou et al., 2006] Dimitracopoulou, A., Bruillard, E., et al. (2006). Enrichir les interfaces de forums par la visualisation d’analyses automatiques des interactions et du contenu. *Revue des Sciences et Technologies de l’Information et de la Communication pour l’Education et la Formation (STICEF)*, 13. (Cité en page 29.)
- [Djouad et al., 2009] Djouad, T., Mille, A., Reffay, C., Benmohamed, M., et al. (2009). Ingénierie des indicateurs d’activités à partir de traces modélisées pour un environnement informatique d’apprentissage humain. *Revue des Sciences et Technologies*

- de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, 16. (Cité en pages 29 et 30.)
- [Djurisic et al., 2004] Djurisic, M., Antic, S., Chen, W. R., and Zecevic, D. (2004). Voltage imaging from dendrites of mitral cells: Epsp attenuation and spike trigger zones. *The Journal of neuroscience*, 24(30):6703–6714. (Cité en page 207.)
- [Dringus and Ellis, 2005] Dringus, L. P. and Ellis, T. (2005). Using data mining as a strategy for assessing asynchronous discussion forums. *Computers & Education*, 45(1):141–160. (Cité en pages 49 et 175.)
- [Duda et al., 2000] Duda, R., Hart, P., and Stork, D. (2000). Pattern classification. 2nd edn wiley. *New York*. (Cité en page 153.)
- [Dutt et al., 2014] Dutt, A., Aghabozrgi, S., Akmal Binti Ismail, M., and Mahrooian, H. (2014). Clustering algorithms applied in educational data mining. *International Journal of Information and Electronics Engineering (IJIEE)*. (Cité en page 257.)
- [Egner et al., 2010] Egner, T., Monti, J. M., and Summerfield, C. (2010). Expectation and surprise determine neural population responses in the ventral visual stream. *The Journal of Neuroscience*, 30(49):16601–16608. (Cité en page 98.)
- [Eigen, 1973] Eigen, M. (1973). *Ingo Rechenberg Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. mit einem Nachwort von Manfred Eigen, Friedrich Frommann Verlag, Struttgart-Bad Cannstatt. (Cité en page 65.)
- [Elman, 1998] Elman, J. L. (1998). *Rethinking innateness: A connectionist perspective on development*, volume 10. MIT press. (Cité en page 89.)
- [Endres, 2006] Endres, D. M. (2006). *Bayesian and information-theoretic tools for neuroscience*. PhD thesis, University of St Andrews. (Cité en page 226.)
- [Erol et al., 1994] Erol, K., Hendler, J., and Nau, D. S. (1994). HTN planning: complexity and expressivity. In *AAAI'94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 2)*, pages 1123–1128. (Cité en page 75.)
- [Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231. (Cité en page 64.)
- [Fayyad et al., 1996a] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996a). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37. (Cité en page 44.)
- [Fayyad et al., 1996b] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., et al. (1996b). Knowledge discovery and data mining: Towards a unifying framework. In *KDD*, volume 96, pages 82–88. (Cité en page 44.)
- [Felleman and Van Essen, 1991] Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1(1):1–47. (Cité en page 89.)
- [Feng et al., 2009] Feng, M., Beck, J. E., and Heffernan, N. T. (2009). Using learning decomposition and bootstrapping with randomization to compare the impact of

- different educational interventions on learning. *International Working Group on Educational Data Mining*. (Cité en page 49.)
- [Feng and Heffernan, 2005] Feng, M. and Heffernan, N. T. (2005). Informing teachers live about student learning: Reporting in the assistent system. In *The 12th Annual Conference on Artificial Intelligence in Education Workshop on Usage Analysis in Learning Systems*. (Cité en page 148.)
- [Festinger, 1954] Festinger, L. (1954). A theory of social comparison processes. *Human relations*, 7(2):117–140. (Cité en page 173.)
- [Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188. (Cité en page 61.)
- [Fix and Hodges Jr, 1951] Fix, E. and Hodges Jr, J. L. (1951). Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, DTIC Document. (Cité en page 62.)
- [Fodor, 1975] Fodor, J. A. (1975). *The Language of Thought*. Cambridge, MA: The MIT Press. (Cité en pages 70 et 71.)
- [Földiák, 1990] Földiák, P. (1990). Forming sparse representations by local anti-hebbian learning. *Biological cybernetics*, 64(2):165–170. (Cité en page 226.)
- [Földiák, 1991] Földiák, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200. (Cité en page 97.)
- [Földiák and Young, 1995] Földiák, P. and Young, M. P. (1995). Sparse coding in the primate cortex. *The handbook of brain theory and neural networks*, 1:1064–1068. (Cité en page 82.)
- [Franklin JL, 1992] Franklin JL, Fickbohm DJ, W. A. (1992). Long-term regulation of neuronal calcium currents by prolonged changes of membrane potential. *Journal of Neuroscience*, pages 1726–35. (Cité en page 213.)
- [Freeman et al., 2014] Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., Jordt, H., and Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences*, page 201319030. (Cité en page 16.)
- [Freund and Schapire, 1995] Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer. (Cité en page 62.)
- [Fritzke et al., 1995] Fritzke, B. et al. (1995). A growing neural gas network learns topologies. *Advances in neural information processing systems*, 7:625–632. (Cité en page 226.)
- [Fukushima, 1988] Fukushima, K. (1988). Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130. (Cité en page 92.)
- [Gallant, 2015] Gallant, J. (2015). Brain reading: Mapping, modeling, and decoding the brain under naturalistic conditions. In *2015 AAAS Annual Meeting (12-16 February 2015)*. aaas. (Cité en page 80.)

- [Ganter et al., 2005] Ganter, B., Stumme, G., and Wille, R. (2005). *Formal concept analysis: foundations and applications*, volume 3626. springer. (Cit  en page 49.)
- [Gaudioso and Talavera, 2006] Gaudioso, E. and Talavera, L. (2006). Data mining to support tutoring in virtual learning communities: Experiences and challenges. *Data Mining in E-Learning(Advances in Management Information)*, 4:207–225. (Cit  en page 32.)
- [Gavornik and Bear, 2014] Gavornik, J. P. and Bear, M. F. (2014). Learned spatio-temporal sequence recognition and prediction in primary visual cortex. *Nature neuroscience*. (Cit  en page 98.)
- [Gedeon and Turner, 1993] Gedeon, T. and Turner, S. (1993). Explaining student grades predicted by a neural network. In *Neural Networks, 1993. IJCNN'93-Nagoya. Proceedings of 1993 International Joint Conference on*, volume 1, pages 609–612. IEEE. (Cit  en page 54.)
- [Gelman et al., 2004] Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). Bayesian data analysis. crc. *Chapman&Hall, Boca Raton, FL*. (Cit  en page 158.)
- [George and Hawkins, 2005] George, D. and Hawkins, J. (2005). A hierarchical bayesian model of invariant pattern recognition in the visual cortex. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 3, pages 1812–1817. IEEE. (Cit  en page 107.)
- [George and Hawkins, 2009] George, D. and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS computational biology*, 5(10):e1000532. (Cit  en pages 102 et 107.)
- [Gibson, 1979a] Gibson, J. (1979a). The ecological approach to visual perception. (Cit  en page 127.)
- [Gibson, 1979b] Gibson, J. J. (1979b). *The ecological approach to visual perception*. Hillsdale ; New Jersey ; London. (Cit  en page 75.)
- [Glorot et al., 2011] Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 513–520. (Cit  en page 95.)
- [Goodman, 1983] Goodman, N. (1983). *Fact, fiction, and forecast*. Harvard University Press. (Cit  en page 92.)
- [Granger, 1969] Granger, C. W. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438. (Cit  en page 63.)
- [Gregory, 1980] Gregory, R. L. (1980). Perceptions as hypotheses. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 290(1038):181–197. (Cit  en page 98.)
- [Gross, 2002] Gross, C. G. (2002). Genealogy of the “grandmother cell”. *The Neuroscientist*, 8(5):512–518. (Cit  en page 203.)

- [Grossberg, 1976] Grossberg, S. (1976). Adaptive pattern classification and universal recoding, i: Parallel development and coding of neural feature detectors & ii: Feedback, expectation, olfaction, and illusions. *Biological Cybernetics*, 23:121–134 & 187–202. (Cité en page 224.)
- [Grossberg and Versace, 2008] Grossberg, S. and Versace, M. (2008). Spikes, synchrony, and attentive learning by laminar thalamocortical circuits. *Brain research*, 1218:278–312. (Cité en page 225.)
- [Hannan and Quinn, 1979] Hannan, E. J. and Quinn, B. G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 190–195. (Cité en page 158.)
- [Harnad, 1990] Harnad, S. (1990). The symbol grounding problem. *Physica*, 42:335–346. (Cité en page 71.)
- [Hart et al., 2008] Hart, D., Goertzel, B., and Llc, N. (2008). Opencog: A software framework for integrative artificial general. In *Intelligence.*” *Proceedings of the First AGI Conference.* (Cité en page 73.)
- [Hartigan, 1972] Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the american statistical association*, 67(337):123–129. (Cité en page 49.)
- [Hawkins and Blakeslee, 2004] Hawkins, J. and Blakeslee, S. (2004). *On Intelligence.* Henry Holt, New York. (Cité en pages 11, 21, 74, 119 et 120.)
- [Hawkins et al., 2009] Hawkins, J., George, D., and Niemasik, J. (2009). Sequence memory for prediction, inference and behaviour. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1203–1209. (Cité en page 107.)
- [Hebb, 1949] Hebb, D. (1949). *The Organisation Of Behavior.* Wiley. (Cité en pages 79, 212 et 220.)
- [Henson et al., 2009] Henson, R. A., Templin, J. L., and Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, 74(2):191–210. (Cité en page 52.)
- [Heraud et al., 2005] Heraud, J.-M., Marty, J.-C., France, L., and Carron, T. (2005). Helping the interpretation of web logs: Application to learning scenario improvement. In *Proceedings of " Usage Analysis in Learning Systems" workshop*, pages 41–48. Citeseer. (Cité en page 29.)
- [Herzog, 2006] Herzog, S. (2006). Estimating student retention and degree-completion time: Decision trees and neural networks vis-à-vis regression. *New Directions for Institutional Research*, 2006(131):17–33. (Cité en page 54.)
- [Hinton et al., 2006] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554. (Cité en page 94.)
- [Hochreiter, 1991] Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen netzen. *Master's thesis, Institut fur Informatik, Technische Universitat, Munchen.* (Cité en page 93.)

- [Hochreiter, 1998] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116. (Cité en page 93.)
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short term memory. *Neural Computation*, 9:1735–1780. (Cité en page 93.)
- [Hodgkin and Huxley, 1952] Hodgkin, A. L. and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of physiology*, 4(117):500–544. (Cité en page 222.)
- [Holland and Reitman, 1978] Holland, J. H. and Reitman, J. S. (1978). Cognitive systems based on adaptive algorithms. In Waterman, D. A. and Hayes-Roth, F., editors, *Pattern directed inference systems*, pages 313–329. Academic Press, New York. (Cité en pages 65 et 128.)
- [Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79:2554–2558. (Cité en page 222.)
- [Hoppe et al., 2003] Hoppe, H. et al. (2003). A web-based tutoring tool with mining facilities to improve learning and teaching. *Artificial intelligence in education: Shaping the future of learning through intelligent technologies*, 97:201. (Cité en page 49.)
- [Horton and Adams, 2005] Horton, J. C. and Adams, D. L. (2005). The cortical column: a structure without a function. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):837–862. (Cité en page 218.)
- [Hubel and Wiesel, 1959] Hubel, D. H. and Wiesel, T. N. (1959). Receptive fields of single neurones in the cat’s striate cortex. *The Journal of physiology*, 148(3):574–591. (Cité en page 92.)
- [Hubel and Wiesel, 1962] Hubel, D. H. and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *The Journal of physiology*, 160(1):106. (Cité en page 92.)
- [Hubel, 1977] Hubel, Wiesel, S. (1977). Orientation columns in macaque monkey visual cortex demonstrated by the 2-deoxyglucose autoradiographic technique. *Nature*, 269:329. (Cité en page 218.)
- [Hurley and Weibelzahl, 2007] Hurley, T. and Weibelzahl, S. (2007). Using motsart to support on-line teachers in student motivation. In *Creating New Learning Experiences on a Global Scale*, pages 101–111. Springer. (Cité en page 49.)
- [Hyvärinen et al., 2009] Hyvärinen, A., Hurri, J., and Hoyer, P. O. (2009). *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.*, volume 39. Springer. (Cité en page 94.)
- [Héguy, 2003] Héguy, O. (2003). *Architecture comportementale pour l’émergence d’activités coopératives en environnement virtuel*. PhD thesis, Université Toulouse 3 Paul Sabatier. (Cité en page 128.)

- [Izenman, 2008] Izenman, A. J. (2008). *Linear Discriminant Analysis*. Springer. (Cité en page 61.)
- [Jaeger, 2001] Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148:34. (Cité en page 224.)
- [Jarsky et al., 2005] Jarsky, T., Roxin, A., Kath, W. L., and Spruston, N. (2005). Conditional dendritic spike propagation following distal synaptic activation of hippocampal ca1 pyramidal neurons. *Nature neuroscience*, 8(12):1667–1676. (Cité en page 209.)
- [Jaynes, 1957] Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, 106(4):620. (Cité en page 156.)
- [Jaynes, 1994] Jaynes, E. T. (1994). *Probability Theory: The Logic of Science*. Cambridge University Press. (Cité en page 155.)
- [Jermann et al., 2001] Jermann, P., Soller, A., and Muehlenbrock, M. (2001). From mirroring to guiding: A review of the state of art technology for supporting collaborative learning. In P. Dillenbourg, A. Eurelings, . K. H., editor, *Proceedings of the European Conference on Computer-Supported Collaborative Learning EuroCSCL-2001. Maastricht, The Netherlands*, pages 324–331, Maastricht, Pays-Bas. (Cité en page 30.)
- [Johnson and Johnson, 1980] Johnson, D. W. and Johnson, R. T. (1980). Integrating handicapped students into the mainstream. *Exceptional children*. (Cité en page 17.)
- [Kassahun and Sommer, 2005] Kassahun, Y. and Sommer, G. (2005). Efficient reinforcement learning through evolutionary acquisition of neural topologies. In *ESANN*, pages 259–266. (Cité en page 228.)
- [Kiani et al., 2007] Kiani, R., Esteky, H., Mirpour, K., and Tanaka, K. (2007). Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *Journal of neurophysiology*, 97(6):4296–4309. (Cité en page 82.)
- [Kiebel et al., 2008] Kiebel, S. J., Garrido, M. I., Moran, R. J., and Friston, K. J. (2008). Dynamic causal modelling for eeg and meg. *Cognitive neurodynamics*, 2(2):121–136. (Cité en page 64.)
- [Koeclinger et al., 2000] Koeclinger, K. R., Corbett, A. T., and Ritter, S. (2000). Carnegie learning’s cognitive tutor “: Summary research results. (Cité en page 126.)
- [Kohavi et al., 1995] Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145. (Cité en page 159.)
- [Kohonen, 1984] Kohonen, T. (1984). *Self-organization and Associative Memory. (3rd edition 1989)*. Springer-Verlag, Berlin. (Cité en pages 64 et 86.)
- [Laflaquière and Prié, 2007] Laflaquière, J. and Prié, Y. (2007). Des traces modélisées, un nouveau support pédagogique ? In *4th annual scientific conference - LORNET*. (Cité en page 21.)

- [Laird, 2012] Laird, J. E. (2012). *The Soar Cognitive Architecture*. The MIT Press. (Cité en page 73.)
- [Lapicque, 1907] Lapicque, L. (1907). Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *J. Physiol. Pathol. Gen*, 9(1):620–635. (Cité en page 221.)
- [Larkum et al., 1999] Larkum, M. E., Zhu, J. J., and Sakmann, B. (1999). A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341. (Cité en page 209.)
- [Lashley, 1929] Lashley, K. S. (1929). *Brain mechanism and intelligence: A quantitative study of injuries to the brain*. Chicago University Press, Chicago. (Cité en page 219.)
- [Lashley, 1950] Lashley, K. S. (1950). In search of the engram. In academic press, editor, *Symposium of the Society for Experimental Biology*, New-York. (Cité en page 219.)
- [Lavallard, 2008] Lavallard, A. (2008). *Exploration interactive d'archives de forums: Le cas des jeux de rôle en ligne*. PhD thesis, Université de Caen. (Cité en page 35.)
- [Lawless and Brown, 1997] Lawless, K. A. and Brown, S. W. (1997). Multimedia learning environments: Issues of learner control and navigation. *Instructional Science*, 25(2):117–131. (Cité en page 53.)
- [Le et al., 2012] Le, Q. V., Monga, R., Devin, M., Chen, K., Corrado, G. S., Dean, J., and Ng, A. Y. (2012). Building high-level features using large scale unsupervised learning. (Cité en page 94.)
- [LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551. (Cité en page 93.)
- [Lee and Mumford, 2003] Lee, T. S. and Mumford, D. (2003). Hierarchical bayesian inference in the visual cortex. *JOSA A*, 20(7):1434–1448. (Cité en page 110.)
- [Lenat, 1995] Lenat, D. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38:33–38. (Cité en page 73.)
- [Liu et al., 2012a] Liu, J., Xu, G., and Ying, Z. (2012a). Data-driven learning of q-matrix. *Applied psychological measurement*, 36(7):548–564. (Cité en page 52.)
- [Liu et al., 2012b] Liu, X., Ramirez, S., Pang, P. T., Puryear, C. B., Govindarajan, A., Deisseroth, K., and Tonegawa, S. (2012b). Optogenetic stimulation of a hippocampal engram activates fear memory recall. *Nature*, 484(7394):381–385. (Cité en page 220.)
- [Loisier, 2011] Loisier, J. (2011). Les nouveaux outils d'apprentissage encouragent-ils réellement la performance et la réussite des étudiants en FAD ? *Document pour le Réseau d'enseignement francophone à distance du Canada*. (Cité en page 16.)
- [London et al., 2010] London, M., Roth, A., Beeren, L., Häusser, M., and Latham, P. E. (2010). Sensitivity to perturbations in vivo implies high noise and suggests rate coding in cortex. *Nature*, 466(7302):123–127. (Cité en page 80.)

- [Lu, 2004] Lu, J. (2004). Personalized e-learning material recommender system. In *Proc. of the Int. Conf. on Information Technology for Application*, pages 374–379. (Cité en page 48.)
- [López et al., 2012] López, M., Luna, J., Romero, C., and Ventura, S. (2012). Classification via clustering for predicting final marks based on student participation in forums. In *Educational Data Mining Proceedings*. (Cité en pages 53 et 54.)
- [Ma et al., 2000] Ma, Y., Liu, B., Wong, C. K., Yu, P. S., and Lee, S. M. (2000). Targeting the right students using data mining. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–464. ACM. (Cité en page 54.)
- [Machine Learning Group at the University of Waikato, 2013] Machine Learning Group at the University of Waikato (2013). Weka 3: Data Mining Software in Java. <http://www.cs.waikato.ac.nz/ml/weka>. (Cité en page 143.)
- [Maletic-Savatic et al., 1999] Maletic-Savatic, M., Malinow, R., and Svoboda, K. (1999). Rapid dendritic morphogenesis in ca1 hippocampal dendrites induced by synaptic activity. *Science*, 283(5409):1923–1927. (Cité en page 214.)
- [Markram, 2006] Markram, H. (2006). The blue brain project. *Nature Reviews Neuroscience*, 7(2):153–160. (Cité en page 113.)
- [Markram et al., 1997] Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic apss and epsps. *Science*, 275(5297):213–215. (Cité en page 213.)
- [Markram and Sakmann, 1995] Markram, H. and Sakmann, B. (1995). Action potentials propagating back into dendrites trigger changes in efficacy of single-axon synapses between layer v pyramidal neurons. In *Soc. Neurosci. Abstr*, volume 21, page 2007. (Cité en page 213.)
- [Martinetz et al., 1991] Martinetz, T., Schulten, K., et al. (1991). *A "neural-gas" network learns topologies*. University of Illinois at Urbana-Champaign. (Cité en pages 64 et 226.)
- [Maxwell and Eigenfeldt, 2008] Maxwell, J. B. and Eigenfeldt, A. (2008). A music database and query system for recombinant composition. In *ISMIR*, pages 75–80. (Cité en page 107.)
- [May et al., 2007] May, M., George, S., Prévôt, P., et al. (2007). Keeping tracks of users' activities on discussion forum: Reflection and proposal. *Learning Through Online Exchanges: Tools, Tasks, Interactions, Multimodality, Corpora (EPAL 2007)*. (Cité en page 35.)
- [Mazza and Milani, 2005] Mazza, R. and Milani, C. (2005). Exploring usage analysis in learning systems: Gaining insights from visualisations. In *AIED'05 workshop on Usage analysis in learning systems*, pages 65–72. (Cité en pages 148 et 149.)
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of mathematical biophysics*, 5:115–133. Reprinted in McCulloch, W. S., *Embodiments of mind*. Cambridge, MA: MIT Press. (Cité en page 82.)

- [McEneaney, 2001] McEneaney, J. E. (2001). Graphic and numerical methods to assess navigation in hypertext. *International Journal of Human-Computer Studies*, 55(5):761–786. (Cité en page 52.)
- [McLachlan, 2004] McLachlan, G. (2004). *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons. (Cité en page 61.)
- [McLean, 1990] McLean, P. (1990). *The Triune Brain in Evolution*. Springer. (Cité en page 199.)
- [Merceron and Yacef, 2008] Merceron, A. and Yacef, K. (2008). Interestingness measures for associations rules in educational data. *EDM*, 8:57–66. (Cité en page 53.)
- [Merceron et al., 2004] Merceron, A., Yacef, K., et al. (2004). Train, store, analyse for more adaptive teaching. In *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie*, pages 52–58. (Cité en pages 35 et 148.)
- [Miller and Lommel, 2006] Miller, J. W. and Lommel, P. H. (2006). Biomimetic sensory abstraction using hierarchical quilted self-organizing maps. In *Optics East 2006*, pages 63840A–63840A. International Society for Optics and Photonics. (Cité en page 98.)
- [Minaei-Bidgoli et al., 2003] Minaei-Bidgoli, B., Kashy, D. A., Kortmeyer, G., and Punch, W. F. (2003). Predicting student performance: an application of data mining methods with an educational web-based system. In *Frontiers in Education, 2003. FIE 2003 33rd Annual*, volume 1, pages T2A–13. IEEE. (Cité en page 54.)
- [Minaei-Bidgoli et al., 2004] Minaei-Bidgoli, B., Tan, P.-N., and Punch, W. F. (2004). Mining interesting contrast rules for a web-based educational system. In *Machine Learning and Applications, 2004. Proceedings. 2004 International Conference on*, pages 320–327. IEEE. (Cité en page 53.)
- [Minsky, 1985] Minsky, M. (1985). *The society of mind*. Simon and Schuster Inc., New York, NY. (Cité en page 72.)
- [Minsky, 1988] Minsky, M. (1988). *Society of mind*. Simon and Schuster. (Cité en page 127.)
- [Minsky and Papert, 1968] Minsky, M. and Papert, S. (1968). *Perceptrons: an Introduction to Computational Geometry*. MIT Press. (Cité en page 86.)
- [Mirkin, 1998] Mirkin, B. (1998). *Mathematical classification and clustering: From how to what and why*. Springer. (Cité en page 49.)
- [Mitchell, 1997] Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill, Boston, MA. (Cité en page 47.)
- [Monk, 2005] Monk, D. (2005). Using data mining for e-learning decision making. *The Electronic Journal of e-Learning*, 3(1):41–54. (Cité en page 53.)
- [Moodle Trust, 2013] Moodle Trust (2013). Moodle official site. <http://moodle.org>. (Cité en page 21.)

- [Moravec, 1988] Moravec, H. (1988). *Mind Children*. Harvard University Press. (Cité en page 72.)
- [Morse et al., 2010] Morse, A. F., De Greeff, J., Belpeame, T., and Cangelosi, A. (2010). Epigenetic robotics architecture (era). *Autonomous Mental Development, IEEE Transactions on*, 2(4):325–339. (Cité en page 99.)
- [Morse and Ziemke, 2009] Morse, A. F. and Ziemke, T. (2009). Action, detection, and perception: A computational model of the relation between movement and orientation selectivity in the cerebral cortex. In *Proceedings of the 31th Annual Conference of the Cognitive Science Society*, pages 585–590. (Cité en page 99.)
- [Mostow et al., 2005] Mostow, J., Beck, J., Cen, H., Cuneo, A., Gouvea, E., and Heiner, C. (2005). An educational data mining tool to browse tutor-student interactions: Time will tell. In *Proceedings of the Workshop on Educational Data Mining, National Conference on Artificial Intelligence*, pages 15–22. (Cité en page 48.)
- [Mountcastle, 1957] Mountcastle, V. B. (1957). Modality and topographic properties of single neurons of cat’s somatic sensory cortex. *J. neurophysiol*, 20(4):408–434. (Cité en page 216.)
- [Mountcastle, 1997] Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain*, 120(4):701–722. (Cité en pages 80, 217 et 218.)
- [Mountcastle, 1998] Mountcastle, V. B. (1998). *Perceptual neuroscience: The cerebral cortex*. Harvard University Press. (Cité en page 208.)
- [Mumford, 1992] Mumford, D. (1992). On the computational architecture of the neocortex. *Biological cybernetics*, 66(3):241–251. (Cité en page 100.)
- [Nebot et al., 2006] Nebot, A., Castro, F., Vellido, A., and Mugica, F. (2006). Identification of fuzzy models to predict students performance in an e-learning environment. In *The Fifth IASTED International Conference on Web-Based Education, WBE*, pages 74–79. (Cité en page 55.)
- [Nguyen et al., 2012] Nguyen, V.-A., Starzyk, J. A., and Goh, W.-B. (2012). Sequence recognition with spatio-temporal long-term memory organization. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE. (Cité en page 97.)
- [Nisbet et al., 2009] Nisbet, R., Elder IV, J., and Miner, G. (2009). *Handbook of statistical analysis and data mining applications*. Academic Press. (Cité en page 45.)
- [Novikoff et al., 2012] Novikoff, T. P., Kleinberg, J. M., and Strogatz, S. H. (2012). Education of a model student. *Proceedings of the National Academy of Sciences*, 109(6):1868–1873. (Cité en page 53.)
- [Numenta, 2007] Numenta (2007). Problems that fit HTMs. Technical report, Numenta. (Cité en page 76.)
- [Numenta, 2010] Numenta (2010). HTM cortical learning algorithms. Technical report, Numenta. (Cité en pages 21, 107 et 120.)

- [Numenta, 2013] Numenta (2013). The science of anomaly detection. Technical report, Numenta. (Cité en pages 120 et 143.)
- [Olshausen et al., 1996] Olshausen, B. A. et al. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609. (Cité en pages 88 et 226.)
- [Olshausen and Field, 2004] Olshausen, B. A. and Field, D. J. (2004). Sparse coding of sensory inputs. *Current opinion in neurobiology*, 14(4):481–487. (Cité en page 82.)
- [Panzoli, 2008] Panzoli, D. (2008). *Proposition de l'architecture Cortexionist pour l'intelligence comportementale de créatures artificielles*. PhD thesis, Université Toulouse 3 - Paul Sabatier. (Cité en pages 22 et 129.)
- [Pardos et al., 2007] Pardos, Z. A., Heffernan, N. T., Anderson, B., and Heffernan, C. L. (2007). The effect of model granularity on student performance prediction using bayesian networks. In *User Modeling 2007*, pages 435–439. Springer. (Cité en page 55.)
- [Pasley et al., 2012] Pasley, B. N., David, S. V., Mesgarani, N., Flinker, A., Shamma, S. A., Crone, N. E., Knight, R. T., and Chang, E. F. (2012). Reconstructing speech from human auditory cortex. *PLoS biology*, 10(1):e1001251. (Cité en page 80.)
- [Paviotti et al., 2012] Paviotti, G., Rossi, P. G., and Zarka, P. (2012). *Intelligent Tutoring Systems: An Overview*. Pensa Multimedia. (Cité en page 126.)
- [Pavlik Jr et al., 2009a] Pavlik Jr, P. I., Cen, H., and Koedinger, K. R. (2009a). Learning factors transfer analysis: Using learning curve analysis to automatically generate domain models. *Online Submission*. (Cité en page 51.)
- [Pavlik Jr et al., 2009b] Pavlik Jr, P. I., Cen, H., and Koedinger, K. R. (2009b). Performance factors analysis—a new alternative to knowledge tracing. In *AIED 2009*. (Cité en page 51.)
- [Pearl, 2000] Pearl, J. (2000). *Causality: models, reasoning and inference*, volume 29. Cambridge Univ Press. (Cité en page 63.)
- [Pennachin and Goertzel, 2007] Pennachin, C. and Goertzel, B. (2007). Contemporary approaches to artificial general intelligence. In *Goertzel and C. Pennachin, Artificial General Intelligence*, pages 1–30. Springer. (Cité en page 72.)
- [Pham Thi Ngoc et al., 2009] Pham Thi Ngoc, D., Iksal, S., Choquet, C., and Klinger, E. (2009). UTL-CL: A Declarative Calculation Language Proposal for a Learning Tracks Analysis Process. In *Proceedings of the 2009 Ninth IEEE International Conference on Advanced Learning Technologies, ICAALT '09*, pages 681–685, Washington, DC, USA. IEEE Computer Society. (Cité en pages 29 et 171.)
- [Pinto et al., 2008] Pinto, N., Cox, D. D., and DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS computational biology*, 4(1):e27. (Cité en page 97.)
- [Pinto and Engel, 2009] Pinto, R. and Engel, P. (2009). Loopsom: a robust som variant using self-organizing temporal feedback connections. In *Proceedings of the 7th ENIA-BRAZILIAN meeting on Artificial Intelligence*. (Cité en page 98.)

- [Powell, 1985] Powell, M. J. (1985). Radial basis functions for multi-variable interpolation: A review. *IMA Conference on Algorithms for the Approximation of Functions and Data*. (Cité en page 221.)
- [Prié, 2006] Prié, A. M.-Y. (2006). Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation/logique de conception. (Cité en page 29.)
- [Putnam, 1963] Putnam, H. (1963). Brains and behavior. In *Analytical Philosophy: Second Series*. Blackwell. (Cité en page 70.)
- [Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106. (Cité en page 62.)
- [Quinlan, 1993] Quinlan, J. R. (1993). *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann. (Cité en page 62.)
- [R Baddeley, 1997] R Baddeley, E. T. R. e. a. (1997). Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proceedings of the Royal Society of London for Biological Sciences*, 264:1775–1783. (Cité en page 82.)
- [Rafferty and Yudelson, 2007] Rafferty, A. N. and Yudelson, M. (2007). Applying learning factors analysis to build stereotypic student models. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS*, 158:697. (Cité en page 51.)
- [Raizada and Grossberg, 2003] Raizada, R. D. and Grossberg, S. (2003). Towards a theory of the laminar architecture of cerebral cortex: Computational clues from the visual system. *Cerebral Cortex*, 13(1):100–113. (Cité en page 225.)
- [Rakic, 1995] Rakic, P. (1995). A small step for the cell, a giant leap for mankind: a hypothesis of neocortical expansion during evolution. *Trends in neurosciences*, 18(9):383–388. (Cité en page 217.)
- [Rakic, 2008] Rakic, P. (2008). Confusing cortical columns. *Proceedings of the National Academy of Sciences*, 105(34):12099–12100. (Cité en page 218.)
- [Ranzato et al., 2007] Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. In *Proceedings of NIPS*. (Cité en page 94.)
- [Rao and Ballard, 1999] Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87. (Cité en page 100.)
- [Rawlinson and Kowadlo, 2012] Rawlinson, D. and Kowadlo, G. (2012). Generating Adaptive Behaviour within a Memory-Prediction Framework. *PLoS ONE*, 7. (Cité en page 128.)
- [Rebane and Pearl, 2013] Rebane, G. and Pearl, J. (2013). The recovery of causal poly-trees from statistical data. *arXiv preprint arXiv:1304.2736*. (Cité en page 63.)
- [Reffay et al., 2006] Reffay, C., Lancieri, L., et al. (2006). Quand l'analyse quantitative fait parler les forums de discussion. *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (STICEF)*, 13. (Cité en page 35.)

- [Reynolds, 1987] Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *ACM SIGGRAPH Computer Graphics*, 21(4):25–34. (Cité en page 127.)
- [Riesenhuber and Poggio, 1999] Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2(11):1019–1025. (Cité en page 93.)
- [Rish, 2001] Rish, I. (2001). An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. (Cité en page 61.)
- [Rizzolatti and Craighero, 2004] Rizzolatti, G. and Craighero, L. (2004). The mirror-neuron system. *Annu. Rev. Neurosci.*, 27:169–192. (Cité en page 203.)
- [Rognetta et al., 2012] Rognetta, J., Jammot, J., and Tardy, F. (2012). *La république des réseaux. Périls et promesses d’une révolution numérique*. Fayard. (Cité en page 13.)
- [Rokach, 2008] Rokach, L. (2008). *Data mining with decision trees: theory and applications*, volume 69. World scientific. (Cité en page 62.)
- [Rolls and Milward, 2000] Rolls, E. T. and Milward, T. (2000). A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition, and information-based performance measures. *Neural Computation*, 12(11):2547–2572. (Cité en page 97.)
- [Rolls and Tovee, 1995] Rolls, E. T. and Tovee, M. J. (1995). Sparseness of the neuronal representation of stimuli in the primate temporal visual cortex. *Journal of Neurophysiology*, 73(2):713–726. (Cité en page 82.)
- [Romero et al., 2009] Romero, C., González, P., Ventura, S., del Jesús, M. J., and Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: A practical application using moodle data. *Expert Systems with Applications*, 36(2):1632–1644. (Cité en page 54.)
- [Romero and Ventura, 2007] Romero, C. and Ventura, S. (2007). Educational data mining: A survey from 1995 to 2005. *Expert systems with applications*, 33(1):135–146. (Cité en page 48.)
- [Romero and Ventura, 2010] Romero, C. and Ventura, S. (2010). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 40(6):601–618. (Cité en page 48.)
- [Romero et al., 2003] Romero, C., Ventura, S., De Bra, P., and De Castro, C. (2003). Discovering prediction rules in aha! courses. In *User Modeling 2003*, pages 25–34. Springer. (Cité en page 49.)
- [Romero et al., 2008] Romero, C., Ventura, S., and García, E. (2008). Data mining in course management systems: Moodle case study and tutorial. *Computers and Education*, pages 368–384. (Cité en page 32.)

- [Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: probabilistic model for information storage and organisation in the brain. *Psychological Review*, 65:386–408. (Cité en pages 62 et 83.)
- [Rubin, 1974] Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688. (Cité en page 63.)
- [Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536. (Cité en pages 62 et 85.)
- [Rummery and Niranjana, 1994] Rummery, G. A. and Niranjana, M. (1994). *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering. (Cité en page 65.)
- [Sanchez, 2004] Sanchez, S. (2004). *Mécanismes évolutionnistes pour la simulation comportementale d'acteurs virtuels*. PhD thesis, Université de Sciences Sociales Toulouse 1. (Cité en page 128.)
- [Sanza, 2001] Sanza, C. (2001). *Evolution d'entités virtuelles coopératives par système de classifieurs*. PhD thesis, Université Toulouse 3 Paul Sabatier. (Cité en page 128.)
- [Schmidhuber, 1992] Schmidhuber, J. (1992). Learning complex, extended sequences using the principle of history compression. *Neural Computation*, 4(2):234–242. (Cité en pages 93 et 101.)
- [Schmidhuber et al., 1993] Schmidhuber, J. H., Mozer, M. C., and Prelinger, D. (1993). Continuous history compression. In *Proc. of Intl. Workshop on Neural Networks, RWTH Aachen*. Citeseer. (Cité en page 101.)
- [Schunk, 2003] Schunk, D. H. (2003). Self-efficacy for reading and writing: Influence of modeling, goal setting, and self-evaluation. *Reading & Writing Quarterly*, 19(2):159–172. (Cité en page 173.)
- [Schwarz et al., 1978] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464. (Cité en page 157.)
- [Scoville and Milner, 1957] Scoville, W. B. and Milner, B. (1957). Loss of recent memory after bilateral hippocampal lesions. *Journal of neurology, neurosurgery, and psychiatry*, 20(1):11. (Cité en page 219.)
- [Searle, 1980] Searle, J. (1980). Minds, brains and programs. In *The Behavioral and Brain Sciences*, volume 3. Cambridge University Press. (Cité en page 71.)
- [Seok and Kim, 2008] Seok, K.-H. and Kim, Y. S. (2008). A new robot motion authoring method using htm. In *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pages 2058–2061. IEEE. (Cité en page 107.)
- [Settouti et al., 2006] Settouti, L. S., Prié, Y., Mille, A., and Marty, J.-C. (2006). Systèmes à base de trace pour l'apprentissage humain. In *Colloque international TICE*. (Cité en pages 29 et 30.)
- [Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 et 623–656. (Cité en pages 70 et 156.)

- [Shannon and Weaver, 1949] Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois. republished in paperback 1963. (Cité en page 70.)
- [Shearer, 2000] Shearer, C. (2000). The crisp-dm model: the new blueprint for data mining. *Journal of data warehousing*, 5(4):13–22. (Cité en page 45.)
- [Sims, 1990] Sims, C. A. (1990). Macroeconomics and reality. *Modelling Economic Series*. Clarendon Press, Oxford. (Cité en page 58.)
- [Sims, 1994] Sims, K. (1994). Evolving 3d morphology and behavior by competition. *Artificial life*, 1(4):353–372. (Cité en page 127.)
- [Smith and Squire, 2009] Smith, C. N. and Squire, L. R. (2009). Medial temporal lobe activity during retrieval of semantic memory is related to the age of the memory. *The journal of Neuroscience*, 29(4):930–938. (Cité en page 220.)
- [Smolensky, 1986] Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. (Cité en page 224.)
- [Soller et al., 2005] Soller, A., Martínez, A., Jermann, P., and Muehlenbrock, M. (2005). From mirroring to guiding: A review of state of the art technology for supporting collaborative learning. *International Journal of Artificial Intelligence in Education*, 15(4):261–290. (Cité en page 30.)
- [Solomonoff, 1964] Solomonoff, R. J. (1964). A formal theory of inductive inference. part i. *Information and control*, 7(1):1–22. (Cité en page 155.)
- [Spiegelhalter et al., 2002] Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639. (Cité en page 158.)
- [Spruston, 2008] Spruston, N. (2008). Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3):206–221. (Cité en page 209.)
- [Srikant and Agrawal, 1996] Srikant, R. and Agrawal, R. (1996). *Mining sequential patterns: Generalizations and performance improvements*. Springer. (Cité en page 64.)
- [Srinivasan et al., 1982] Srinivasan, M. V., Laughlin, S. B., and Dubs, A. (1982). Predictive coding: a fresh view of inhibition in the retina. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 216(1205):427–459. (Cité en page 100.)
- [Stanley et al., 2009] Stanley, K. O., D’Ambrosio, D. B., and Gauci, J. (2009). A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212. (Cité en page 228.)
- [Stanley and Miikkulainen, 2002] Stanley, K. O. and Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127. (Cité en page 228.)
- [Stein et al., 2005] Stein, R. B., Gossen, E. R., and Jones, K. E. (2005). Neuronal variability: noise or part of the signal? *Nature Reviews Neuroscience*, 6(5):389–397. (Cité en page 80.)

- [Steinhaus, 1957] Steinhaus, H. (1957). Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci.* (Cit  en page 64.)
- [Stoner et al., 2014] Stoner, R., Chow, M. L., Boyle, M. P., Sunkin, S. M., Mouton, P. R., Roy, S., Wynshaw-Boris, A., Colamarino, S. A., Lein, E. S., and Courchesne, E. (2014). Patches of disorganization in the neocortex of children with autism. *New England Journal of Medicine*, 370(13):1209–1219. (Cit  en page 216.)
- [Stringer and Rolls, 2002] Stringer, S. M. and Rolls, E. T. (2002). Invariant object recognition in the visual system with novel views of 3d objects. *Neural Computation*, 14(11):2585–2596. (Cit  en page 97.)
- [Stuart et al., 2007] Stuart, G., Spruston, N., and Hussler, M. (2007). *Dendrites*. Oxford University Press. (Cit  en page 204.)
- [Sun, 1999] Sun, R. (1999). Artificial intelligence: Connectionist and symbolic approaches. (Cit  en page 72.)
- [Superby et al., 2006] Superby, J.-F., Vandamme, J., and Meskens, N. (2006). Determination of factors influencing the achievement of the first-year university students using data mining methods. In *Workshop on Educational Data Mining*, pages 37–44. (Cit  en page 54.)
- [Sutton, 1988] Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44. (Cit  en page 65.)
- [Sutton and Barto, 1998] Sutton, R. S. and Barto, A. G. (1998). *Introduction to reinforcement learning*. MIT Press. (Cit  en page 60.)
- [Swarup et al., 2005a] Swarup, S., Lakkaraju, K., Klementiev, A., Ray, S. R., Sassen, E., and Munchen, G. (2005a). An anticipatory self-organized map for robust recognition. In *Artificial Intelligence and Applications*, pages 225–229. (Cit  en page 98.)
- [Swarup et al., 2005b] Swarup, S., Lakkaraju, K., Smith, N., and Ray, S. (2005b). A recurrent anticipatory self-organized map for robust time series prediction. In *The Workshop on Self-Organizing Maps (WSOM), Paris, France*. (Cit  en page 98.)
- [Szegedy et al., 2013] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*. (Cit  en page 95.)
- [Talavera and Gaudioso, 2004] Talavera, L. and Gaudioso, E. (2004). Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In *Workshop on artificial intelligence in CSCL. 16th European conference on artificial intelligence*, pages 17–23. (Cit  en page 53.)
- [Tang and McCalla, 2002] Tang, T. Y. and McCalla, G. (2002). Student modeling for a web-based learning environment: a data mining approach. In *AAAI/IAAI*, pages 967–968. (Cit  en page 53.)
- [Tang and McCalla, 2003] Tang, T. Y. and McCalla, G. (2003). Smart recommendation for an evolving e-learning system. In *Workshop on Technologies for Electronic*

- Documents for Supporting Learning, International Conference on Artificial Intelligence in Education (AIED)*, page 2003. (Cité en pages 48 et 53.)
- [Tanguay et al., 2011] Tanguay, V., Gaudreault-Perron, J., Roy, R., and Laferrière, T. (2011). Génération C : les “ C “ en tant qu’étudiants. Technical report, Centre facilitant la recherche et l’innovation dans les organisations. (Cité en page 13.)
- [Tatsuoka, 1983] Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354. (Cité en page 52.)
- [Taylor, 2002] Taylor, J. (2002). Teaching and Learning Online: The Workers, The Lurkers and The Shirkers. *Journal of Chinese Distance Education*, (9):31–37. (Cité en pages 53 et 164.)
- [Tchounikine, 2002] Tchounikine, P. (2002). Quelques éléments sur la conception et l’ingénierie des EIAH. *Information Interaction Intelligence (Actes des 2ème assises nationales du GdR I3, Nancy, décembre 2002)*. (Cité en page 13.)
- [Thorndike, 1898] Thorndike, E. L. (1898). Animal intelligence: An experimental study of the associative processes in animals. *Psychological Monographs: General and Applied*, 2(4):i–109. (Cité en page 220.)
- [Thornton et al., 2008] Thornton, J., Faichney, J., Blumenstein, M., and Hine, T. (2008). Character recognition using hierarchical vector quantization and temporal pooling. In *AI 2008: Advances in Artificial Intelligence*, pages 562–572. Springer. (Cité en page 107.)
- [Thornton et al., 2006] Thornton, J., Gustafsson, T., Blumenstein, M., and Hine, T. (2006). Robust character recognition using a hierarchical bayesian network. In *AI 2006: Advances in Artificial Intelligence*, pages 1259–1264. Springer. (Cité en page 107.)
- [Thorpe, 1990] Thorpe, S. J. (1990). Spike arrival times: A highly efficient coding scheme for neural networks. *Parallel processing in neural systems*, pages 91–94. (Cité en pages 80 et 228.)
- [Thorpe, 2009] Thorpe, S. J. (2009). The speed of categorization in the human visual system. *Neuron*, 62(2):168–170. (Cité en page 101.)
- [Tian et al., 2008] Tian, F., Wang, S., Zheng, C., and Zheng, Q. (2008). Research on e-learner personality grouping based on fuzzy clustering analysis. In *Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on*, pages 1035–1040. IEEE. (Cité en page 53.)
- [Towne et al., 2002] Towne, L., Shavelson, R. J., et al. (2002). *Scientific research in education*. National Academies Press. (Cité en page 48.)
- [Tribus, 1961] Tribus, M. (1961). *Thermostatitics and thermodynamics*. Center for Advanced Engineering Study, Massachusetts Institute of Technology. (Cité en page 156.)
- [Tribus, 1969] Tribus, M. (1969). *Rational descriptions, decisions and designs*. (Cité en page 155.)

- [Tseng et al., 2008] Tseng, J. C., Chu, H.-C., Hwang, G.-J., and Tsai, C.-C. (2008). Development of an adaptive learning system with two sources of personalization information. *Computers & Education*, 51(2):776–786. (Cité en page 20.)
- [Tsunoda, 1985] Tsunoda, T. (1985). *The Japanese Brain: Uniqueness and Universality*. Taishukan. (Cité en page 203.)
- [Turing, 1950] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59:433–460. (Cité en page 70.)
- [Turrigiano et al., 1998] Turrigiano, G. G., Leslie, K. R., Desai, N. S., Rutherford, L. C., and Nelson, S. B. (1998). Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892–896. (Cité en page 214.)
- [Ullman and Bart, 2004] Ullman, S. and Bart, E. (2004). Recognition invariance obtained by extended and invariant features. *Neural Networks*, 17(5):833–848. (Cité en page 97.)
- [van Doremalen and Boves, 2008] van Doremalen, J. and Boves, L. (2008). Spoken digit recognition using a hierarchical temporal memory. In *INTERSPEECH 2008, 9th Annual Conference of the International Speech Communication Association, Brisbane, Australia, September 22-26, 2008*, pages 2566–2569. (Cité en page 107.)
- [VanElzakker et al., 2008] VanElzakker, M., Fevurly, R. D., Breindel, T., and Spencer, R. L. (2008). Environmental novelty is associated with a selective increase in fos expression in the output elements of the hippocampal formation and the perirhinal cortex. *Learning & Memory*, 15(12):899–908. (Cité en page 219.)
- [Vermeulen et al., 2014] Vermeulen, A. E. et al. (2014). *Improving Student Engagement Through Visualization of Course Activities*. PhD thesis. (Cité en pages 149, 150, 173 et 257.)
- [Viau, 2005] Viau, R. (2005). 12 questions sur l'état de la recherche scientifique sur l'impact des TIC sur la motivation à apprendre. *Université de Sherbrooke*. (Cité en page 16.)
- [Villaverde et al., 2006] Villaverde, J. E., Godoy, D., and Amandi, A. (2006). Learning styles' recognition in e-learning environments with feed-forward neural networks. *Journal of Computer Assisted Learning*, 22(3):197–206. (Cité en page 55.)
- [Viterbi, 1967] Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269. (Cité en page 63.)
- [Vuong et al., 2011] Vuong, A., Nixon, T., and Towle, B. (2011). A method for finding prerequisites within a curriculum. In *EDM*, pages 211–216. (Cité en page 48.)
- [Wallis and Rolls, 1997] Wallis, G. and Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Progress in neurobiology*, 51(2):167–194. (Cité en page 97.)
- [Wang, 2000] Wang, P. (2000). Non-axiomatic reasoning system (version 4.1). Technical report, Indiana University, Center for. (Cité en page 73.)

- [Wang, 2005] Wang, Y.-Z. (2005). A ga-based methodology to determine an optimal curriculum for schools. *Expert Systems with Applications*, 28(1):163–174. (Cité en page 48.)
- [Watkins, 1989] Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge. (Cité en page 65.)
- [Welch, 2003] Welch, L. R. (2003). Hidden markov models and the baum-welch algorithm. *IEEE Information Theory Society Newsletter*, 53(4):10–13. (Cité en page 63.)
- [Widrow and Hoff, 1960] Widrow, B. and Hoff, M. T. (1960). Adaptive switching circuits. *IRE WESCON Convention Record*, pages 96–104. (Cité en page 85.)
- [Wolpert and Macready, 1997] Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82. (Cité en page 74.)
- [Wright, 1921] Wright, S. (1921). Correlation and causation. *Journal of agricultural research*, 20(7):557–585. (Cité en page 63.)
- [Xu and Mostow, 2011] Xu, Y. and Mostow, J. (2011). Using logistic regression to trace multiple sub-skills in a dynamic bayes net. In *EDM*, pages 241–246. Citeseer. (Cité en page 50.)
- [Zaki, 2000] Zaki, M. J. (2000). Scalable algorithms for association mining. *Knowledge and Data Engineering, IEEE Transactions on*, 12(3):372–390. (Cité en page 64.)
- [Zakrzewska, 2008] Zakrzewska, D. (2008). Cluster analysis for users’ modeling in intelligent e-learning systems. In *New Frontiers in Applied Artificial Intelligence*, pages 209–214. Springer. (Cité en page 53.)
- [Zang and Lin, 2003] Zang, W. and Lin, F. (2003). Investigation of web-based teaching and learning by boosting algorithms. In *Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on*, pages 445–449. IEEE. (Cité en page 54.)
- [Zaïane, 2002] Zaïane, O. R. (2002). Building a recommender agent for e-learning systems. In *Computers in Education, 2002. Proceedings. International Conference on*, pages 55–59. IEEE. (Cité en page 48.)
- [Zaïane and Luo, 2001] Zaïane, O. R. and Luo, J. (2001). Web usage mining for a better web-based learning environment. In *Proceedings of conference on advanced technology for education*, pages 60–64. Citeseer. (Cité en page 47.)
- [Zhang et al., 2007] Zhang, H., Almeroth, K., Knight, A., Bulger, M., and Mayer, R. (2007). Moodog: Tracking students’ online learning activities. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, volume 2007, pages 4415–4422. (Cité en page 149.)
- [Zhang et al., 2008] Zhang, L., Liu, X., and Liu, X. (2008). Personalized instructing recommendation system based on web mining. In *Young Computer Scientists*,

2008. *ICYCS 2008. The 9th International Conference for*, pages 2517–2521. IEEE.
(Cité en page 49.)
- [Zoccolan et al., 2007] Zoccolan, D., Kouh, M., Poggio, T., and DiCarlo, J. J. (2007).
Trade-off between object selectivity and tolerance in monkey inferotemporal cortex.
The Journal of Neuroscience, 27(45):12292–12307. (Cité en page 81.)

Table des figures

1.1	Le taux de succès des élèves en apprentissage actif ou passif	17
1.2	Notre proposition d'architecture de données pour GIGA	23
2.1	Les traces enregistrées par Moodle	26
2.2	Architecture d'un outil de calcul d'indicateurs	30
2.3	Les indicateurs proposés	39
3.1	Le processus d'exploration de données	45
3.2	La méthodologie CRISP-DM	46
4.1	Schéma d'un neurone formel	83
4.2	Le Perceptron employé pour la vision	84
4.3	Un réseau de neurones multicouche	84
4.4	Un exemple de réseau de neurones récurrent	85
4.5	Classement de couleurs par une carte auto-adaptative	87
4.6	Cartographie hiérarchique des aires du cortex des primates établie par Felleman et Van Essen	90
4.7	Neocognitron	93
4.8	Image très difficile à comprendre initialement, représentant un dalmatien dans un environnement très bruité	100
4.9	Circuit ascendant de l'information	103
4.10	Circuit descendant de l'information	104
4.11	Rôle du thalamus selon Hawkins	104
4.12	Hierarchie de régions HTM	106
4.13	Une région HTM-CLA	108
4.14	Prédiction au sein d'une région HTM-CLA	110
4.15	Représentation d'un filtre de Bloom	112
4.16	Entrées de HTM	114
4.17	Sorties de HTM	116
4.18	Un exemple de convertisseur scalaire pour des entiers allant de 1 à 6	122
4.19	Notre interface de visualisation et de contrôle de simulation HTM	124
4.20	L'architecture Cortexionist	130
4.21	L'environnement de test pour Cortexionist	131

4.22	Quelques exemples des problèmes Maze et Woods. T représente un arbre et F la nourriture (des mots anglais <i>tree</i> et <i>food</i>).	133
4.23	Notre interface de visualisation des problèmes de type Maze/Woods .	136
5.1	Notre proposition d'architecture fonctionnelle pour GIGA	140
5.2	Notre proposition d'architecture applicative pour GIGA	141
5.3	Notre proposition d'architecture technique pour GIGA	142
5.4	Notre proposition d'architecture de données pour GIGA	144
5.5	La page d'accueil de GIGA, centrée sur une formation dont on voit l'IGA	148
5.6	Les différents indicateurs et leur valeur pour tous les étudiants de la formation sélectionnée	149
5.7	L'interface de GISMO montre l'activité des élèves en fonction du temps	150
5.8	L'interface de MooDog fournit agrégation de données et visualisation	151
5.9	L'interface de ProgressTracker fournit agrégation de données et visualisation	152
5.10	Les questionnaires de satisfaction des étudiants utilisant ProgressTracker	153
5.11	Le phénomène du sur-apprentissage	154
5.12	La méthode du coude illustrée	161
5.13	Nombre optimal de groupes trouvé par chaque algorithme pour nos trois formations de test	162
5.14	Nombre optimal de groupes trouvé par chaque algorithme pour nos trois formations de test (suite)	163
5.15	Un exemple de résultat de partitionnement ne présentant que des différences quantitatives	165
A.1	La page d'accueil de GIGA, centrée sur une formation dont on voit l'IGA	177
A.2	Les différents indicateurs et leur valeur pour tous les étudiants de la formation sélectionnée	178
A.3	Les différents indicateurs thématiques	178
A.4	Les différentes statistiques relevant du thème de l'étude de la présence en ligne	179
A.5	Les différentes statistiques relevant du thème de l'étude des contenus	179
A.6	Les moyennes des notes obtenues par l'élève	180
A.7	Les différentes statistiques relevant du thème de l'étude de l'interaction sociale	180
A.8	La participation aux activités notées	181
A.9	Le détail de l'activité d'un élève en lecture de forum	181
A.10	Le détail de l'activité d'un élève en notes obtenues	182
A.11	Le détail de l'activité d'un élève en leçons lues sous forme de PDF .	182
A.12	Le détail de l'activité d'un élève en activités SCORM réalisées	183
A.13	Le détail de l'activité de connexion d'un élève	183
A.14	Le profil d'un élève, regroupant tous ses indicateurs	184

B.1	Diagramme de classes simplifié de HTM-CLA	186
B.2	Diagramme de classes détaillé de HTM-CLA	187
B.3	Diagramme simplifié des classes permettant les fonctionnalités d'apprentissage automatique	188
B.4	Diagramme détaillés des classes permettant les fonctionnalités d'apprentissage automatique	189
B.5	Le diagramme de classes de l'ajout de comportement à HTM	190
B.6	Le diagramme de classes des implémentations Maze et pierre-feuille-ciseaux	191
B.7	Les paquets principaux constituant notre application	191
B.8	Le diagramme de classes des entités de GIGA	192
B.9	Le diagramme de classes des entités liées aux indicateurs	193
B.10	Le diagramme de classes simplifié de l'intégration de Weka à GIGA	194
B.11	Le diagramme de classes détaillé de l'intégration de Weka à GIGA	195
B.12	Diagramme de classes simplifié d'intégration de HTM-CLA à GIGA	196
B.13	Diagramme de classes détaillé d'intégration de HTM-CLA à GIGA	196
C.1	Cellules gliales	199
C.2	Vue en coupe des principales structures cérébrales	200
C.3	Ganglions de la base	200
C.4	Cerveau limbique	201
C.5	Les principales aires corticales	202
C.6	Les homoncules somesthésique et moteur	204
C.7	Schéma d'un neurone	205
C.8	Potentiel d'action d'un neurone	206
C.9	Potentiels post-synaptiques excitatoires suffisant pour déclencher un potentiel d'action	207
C.10	Potentiel post-synaptique excitatoire insuffisant pour déclencher un potentiel d'action	207
C.11	Polarité des neurones	208
C.12	Types de neurones	210
C.13	Synapse chimique	210
C.14	Plasticité cérébrale	214
C.15	Couches corticales et leur population de neurones. On observera aussi la forte connectivité verticale qui est à l'origine du concept de colonnes.	216
D.1	Diverses fonctions d'activation possibles : fonction de Heaviside, fonction signe, fonction linéaire à seuil, fonction sigmoïde.	222
D.2	Le réseau de Hopfield	223
D.3	Une machine de Boltzmann restreinte	224
D.4	Un Echo state network	225
D.5	Les deux modules de l'ART	226
D.6	Apprentissage par un gaz neuronal et un gaz neuronal croissant	227