

Hugues Bersini
Ken Hasselmann

L'intelligence artificielle en pratique avec Python

Recherche, optimisation, apprentissage

2^e édition



● Éditions
EYROLLES

CHEZ LE MÊME ÉDITEUR

SUR LE MÊME THÈME

Y. BENZAKI. – **Les data sciences en 100 questions/réponses.**

N°67951, 2020, 126 pages.

M-R. AMINI. ET AL. – **Data science – Cours et exercices.**

N°67410, 2018, 272 pages.

H. WICKHAM, G. GROLEMUND. – **R pour les data sciences.**

N°67571, 2018, 496 pages.

F. PROVOST, T. FAWCETT. – **Data science pour l'entreprise.**

N°67570, 2018, 384 pages.

E. BIERNAT, M. LUTZ. – **Data science : fondamentaux et études de cas.**

N°14243, 2015, 312 pages.

DANS LA MÊME COLLECTION

B. BAILLY. ET AL. – **Les réseaux 5G.**

N°67898, 2020, 580 pages.

K. NOVAK. – **Administration Linux par la pratique – Tome 2.**

N°67949, 2020, 418 pages.

C. DELANNOY. – **Le guide complet du langage C.**

N°67922, 2020, 876 pages.

J. LORIAUX, T. DEFOSSEZ. – **Emailing : développement et intégration.**

N°67849, 2020, 160 pages.

B. BARRÉ. – **Concevez des applications mobiles avec React Native.**

N°67889, 2^e édition, 2019, 224 pages.

S. RINGUEDÉ. – **SAS.**

N°67631, 4^e édition, 2019, 688 pages.

C. BLAESS. – **Développement système sous Linux.**

N°67760, 5^e édition, 2019, 1080 pages.

Retrouvez nos bundles (livres papier + e-book) et livres numériques sur
<http://izibook.eyrolles.com>

Hugues Bersini
Ken Hasselmann

L'intelligence artificielle en pratique avec Python

Recherche, optimisation, apprentissage

2^e édition

● Éditions
EYROLLES

ÉDITIONS EYROLLES
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre français d'exploitation du droit de copie, 20, rue des Grands Augustins, 75006 Paris.

© Éditions Eyrolles, 2022, 2023 pour la présente édition, ISBN : 978-2-416-01094-1

Avant-propos

Un peu de cinéma

Dans le troisième opus de la série *Die Hard*, le héros interprété par Bruce Willis doit résoudre un ensemble de problèmes logiques, chaque fois en une minute maximum, sinon un horrible méchant risque de faire exploser New York. L'un de ces problèmes implique deux cruches non graduées, de capacité cinq litres et trois litres, que notre héros peut remplir et vider à une fontaine à proximité. Il lui est demandé de remplir en moins d'une minute la cruche de cinq litres avec exactement quatre litres. C'est un problème idéal pour l'Intelligence Artificielle qui l'aurait, elle, résolu en quelques millisecondes. Bruce Willis anticipe plusieurs actions dans sa tête, même plusieurs séquences d'actions, avant de trouver la bonne solution deux secondes avant l'apocalypse. Il faut remplir la cruche de cinq litres avec celle de trois litres, il reste deux litres vides dans la première. On recommence la même opération avec la cruche de trois litres, il reste cette fois un litre dans la cruche de trois litres. Il suffit alors de vider la cruche de cinq litres, la remplir d'abord avec le reste de la cruche de trois litres, le litre restant, et recommencer l'opération pour obtenir les quatre litres dans la cruche de cinq. Pour l'IA, il suffit de décrire l'état du problème en tant que (x, y) , contenu des deux cruches, et coder chaque action possible sous la forme avant→après, par exemple $(0, 0) \rightarrow (5, 0)$ (remplissage de la première) ou $(x, y) \rightarrow (5, y - (5 - x))$, remplissage de la première avec une partie de la seconde. L'IA peut alors facilement enchaîner les actions et trouver la séquence optimale : $(0, 0)$ puis $(0, 3)$ puis $(3, 0)$ puis $(3, 3)$ puis $(5, 1)$ puis $(0, 1)$ puis $(1, 0)$ puis $(1, 3)$ puis $(4, 0)$. Bruce Willis, doté d'un QI bien supérieur à la moyenne, raisonne exactement comme l'IA d'avant, celle des années 1950-1960, celle qui a fait de notre raison la cible à abattre.

Autre film, autre ambiance. Dans ces scènes de cinéma que Steven Spielberg prend tant de plaisir à réaliser, les petits garçons n'ont jamais aucun mal à reconnaître leur maman, que ce soit le héros de *L'empire du soleil*, surtout à la fin du film, ou même le petit robot de *AI* (ça tombe bien)¹. Il leur suffit de percevoir, même de très loin, même dans l'obscurité, même dans une foule, surtout dans une foule, le visage de leur maman, pour que le leur s'illumine. Cette reconnaissance est comme une évidence, un réflexe de survie qui ne nécessite pas le moindre raisonnement, contrairement à celui extrait du film précédent. Il s'agit d'un tout autre type de processus cognitif, bien plus primitif, dont sont dotés les enfants autant que les animaux. Même les singes les plus proches de nous sont totalement inca-

pables de résoudre le problème de *Die Hard*, alors que la reconnaissance d'un visage de primate ou d'humain ne leur demande aucun effort, une dépense minimale de Watts neuronaux. Pour l'IA en revanche, il s'agit là d'un tout autre type de processus cognitifs à reproduire.

Alors que les premiers développements de l'IA cherchaient déjà à reproduire l'ensemble de tous les processus cognitifs humains, ces multiples intelligences, de celles qui nous permettent de raisonner, mais aussi voir, se mouvoir, entendre, parler, écrire, créer... les premières réussites logicielles tinrent surtout à la reproduction de la partie la plus cognitive de nos cerveaux : raisonnement, résolution de problème, maîtrise du langage naturel. On négligea la vision ou la robotique, car supposées beaucoup plus à portée que le jeu d'échec ou le résumé d'un texte (c'est d'une telle simplicité pour nous). Erreur de jugement fatale ! Il fallut attendre de nombreuses années pour que les idées déjà en ébauche aux origines de l'IA (comme les réseaux de neurones) finissent par porter leurs fruits et atteindre les résultats espérés, amenant cette autre tradition de l'IA (ancrée dans nos processus inconscients) à voler la vedette à la version d'origine (ancrée dans nos processus les plus conscients et les plus rationnels). Cette inversion de tendance incite certains chercheurs à aller jusqu'à déclarer que toute la tradition plus cognitive de l'IA, plus logique, plus séquentielle, basée sur la construction d'arbres de raisonnement, et exploitant les connaissances humaines pour simplifier la recherche, l'a conduite dans une impasse².

Dans ce livre, nous nous inscrivons en faux et, parmi nos missions premières, nous souhaitons réhabiliter ces premiers algorithmes de l'IA, de ceux que l'on retrouve abondamment dans les entreprises aujourd'hui, comme lorsque vous faites une recherche sur Google, jouez aux échecs, utilisez Waze ou Uber. Il est vrai que trois facteurs conjoncturels importants sont à l'origine du succès de ces nouveaux algorithmes d'IA bien davantage axés sur l'apprentissage : d'abord quelques innovations algorithmiques (de celles que l'on trouve par exemple dans les architectures plus récentes des réseaux de neurones), mais surtout une quantité extraordinaire de données, comme autant de témoignages du génie humain dont les algorithmes d'apprentissage peuvent s'inspirer même sans les comprendre, et finalement des processeurs informatiques qui se sont extraordinairement « musclés » en puissance de calcul, recourant notamment à de plus en plus de parallélisme (ainsi l'ascendance des GPU sur les CPU pour les réseaux de neurones profonds). Nous en parlerons également.

-
1. Alors qu'il cogitait à la réalisation de son film *Artificial Intelligence*, inspiré de la nouvelle *Les supertoyos durent tout l'été* de Brian Aldiss, un des problèmes taraudant considérablement Stanley Kubrick, connu pour son perfectionnisme maladif, était l'apparence du petit garçon robot, David, le protagoniste du film. Lorsque Spielberg hérita de ce projet à la mort de Kubrick, il ne s'encombra pas de difficultés techniques outre mesure et choisit Haley Joel Osment, qui s'efforça, par la mécanique de sa gestuelle et son statisme, de jouer au robot. Dans le film, une famille en mal d'enfants se voit confier David, représentant d'une nouvelle génération robotique frôlant la perfection et qui, alors qu'il est programmé pour être capable d'un véritable amour filial pour ses parents, surtout sa mère, souffre de l'absence de réciprocité de cette dernière, qui ne parvient pas à s'affranchir de la nature siliconée de son petit. On retrouve le problème relationnel parents/enfants si cher à Spielberg, rendu plus aigu encore par la difficulté dans laquelle se trouve la mère. Elle peine à voir dans David autre chose qu'un toasteur, capable de quelques expressions touchantes mais sans plus, et cela malgré les effets appuyés de séduction déployés à tout venant par le petit robot.
 2. Afin de prendre position dans ce débat, nous recommandons la lecture de *Rebooting Artificial Intelligence we can trust* de Gary Marcus et Ernest Davis ou *Architects of intelligence, the truth about AI from the people building it* de Martin Ford.

Contenu et lectorat du livre

Plusieurs algorithmes d'IA, parmi les plus fameux et les plus usités, vont vous être illustrés, décorqués et présentés dans ce livre. Il s'agit d'algorithmes de recherche (A*), d'optimisation (algorithmes génétiques) et d'apprentissage (renforcement et classification, à partir de réseaux de neurones ou pas). Ils le seront à travers des applications aussi simples que populaires et très certainement connues par la plupart d'entre vous. Nous décrirons d'abord l'algorithme de résolution de manière conceptuelle, puis aborderons plus en détail certaines parties du code permettant à cet algorithme de s'exécuter pour les applications en question. Les codes seront accessibles dans leur intégralité sur un site web dédié et donc récupérables et exécutables en l'état. Nous nous intéresserons également à certaines techniques propres au traitement du langage naturel (comme l'algorithme LDA) et au traitement d'image (comme les filtres convolutifs), deux domaines applicatifs qui ont vécu des améliorations assez conséquentes ces vingt dernières années.

Nous avons fait le choix du langage de programmation Python pour l'écriture et la présentation des codes. Ce choix pourrait sembler paradoxal quand vous lirez souvent dans l'ouvrage qu'un des critères les plus importants dans le choix d'un algorithme s'avère le temps de calcul que son exécution réclame. En effet, Python étant un langage interprété, ayant privilégié par ailleurs la simplicité d'écriture sur la vitesse d'exécution, il n'est nullement recommandé pour celle-ci. C'est bien évidemment cette simplicité d'usage qui en fait depuis quelques années le langage de prédilection pour l'enseignement de l'informatique et de la programmation ; donc, assez logiquement, pour l'enseignement des algorithmes d'IA. Nous ne cherchons donc pas à déroger à cette tradition pédagogique, dont une des conséquences bénéfiques premières est l'extraordinaire offre logicielle sur le Web, notamment pour résoudre les mêmes applications que nous présenterons dans la suite de l'ouvrage. Lorsque nous comparons plusieurs algorithmes quant à leur vitesse d'exécution, nous passons évidemment totalement sous silence le langage de programmation utilisé. Libre à chacun d'entre vous, une fois comprise et assimilée la logique qui se cache derrière l'algorithme, de le transposer dans votre langage de programmation favori, afin de récupérer tout ce temps d'exécution perdu dont Python pourrait plaider coupable. Nous souhaitons aussi le plus souvent révéler la logique algorithmique (codée en Python) qui se dissimule derrière les performances des logiciels, afin de décourager le lecteur de se ruer sans discernement sur les bibliothèques qui implémentent l'algorithme et facilitent son usage, mais en en dissimulant ce cœur algorithmique. La compréhension profonde de ce dernier nous intéresse bien davantage que ses performances. C'est la pédagogie de l'IA qui nous importe, plus que son exploitation empressée.

Ce livre s'adresse à tous les étudiants, en informatique ou pas, qui découvrent l'IA dans leur parcours académique, mais aussi à tous les informaticiens, même les plus confirmés, qui se sentent de plus en plus décontenancés devant l'offre pléthorique des recettes d'IA dont ils n'arrivent pas toujours à comprendre « qui fait quoi ». Il est la conséquence de trente années d'enseignement de l'IA et des milliers d'étudiants que nous avons vu effectuer très souvent le mauvais choix algorithmique pour le problème qu'ils étaient censés résoudre. Parmi les raisons de ces errements : un manque de recul et de questionnement sur le problème, l'absence d'une connaissance plus en profondeur des recettes de l'IA (y compris les plus anciennes), le panurgisme et la facilité d'accès aux recettes les plus récentes et les plus à la mode, tel l'apprentissage profond. Les apprenants pourront, après

avoir assimilé la logique algorithmique, facilement télécharger et exécuter les codes Python, quitte à les remanier, les améliorer et les transposer pour un problème du même acabit.

Remerciements

Hugues Bersini souhaite vivement remercier son co-auteur, Ken Hasselmann, dont il a eu la grande chance de bénéficier de l'accompagnement toutes ces dernières années dans son enseignement de l'IA. Il souhaite aussi remercier tous les membres historiques et ceux toujours présents du laboratoire IRIDIA, qu'il dirige avec son ami et formidable chercheur Marco Dorigo. Il s'agit de centaines et de centaines de chercheurs qui ont fait la réputation d'un laboratoire devenu un incontournable de l'IA dans le monde entier. Il souhaite remercier les milliers d'étudiants qu'il espère avoir sensibilisés, même plus, passionnés à l'IA, surtout ces deux dernières années, que la crise de Covid a rendu nettement plus malaisées, tellement moins agréables pour eux. Finalement, c'est à toute l'équipe d'Eyrolles, avec laquelle il contribue depuis tant d'années et qui continue à le soutenir et l'accompagner avec grand professionnalisme, qu'il souhaite adresser sa plus sincère reconnaissance. Rien ne fait mieux ressortir des lignes de code que la page blanche sur laquelle elles s'écrivent. Et combien de fois lui a-t-il été donné de vivre un eureka salvateur, en feuilletant les pages d'un livre d'informatique ou de programmation, dans un métro, un train, à l'arrière d'une voiture ou faiblement éclairé par une lampe de chevet la tête posée sur l'oreiller. Le livre survivra, même dans sa version papier, surtout dans sa version papier, grâce aux éditeurs qui le défendent avec tant de passion.

Ken Hasselmann tient aussi à remercier Hugues Bersini, pour sa confiance et pour lui avoir offert l'opportunité de participer à l'élaboration de cet ouvrage, et toute l'équipe d'Eyrolles pour l'avoir rendu possible. Il tient à vivement remercier toute l'équipe d'IRIDIA pour l'ambiance chaleureuse et familiale qui règne au sein du laboratoire ! Pour finir, il souhaite remercier ses amis et sa famille pour leur soutien indéfectible et leur amour inconditionnel.

Table des matières

Introduction	1
Installation de Python	5
Installation de poetry	6
Sous Windows	6
Sous Linux	7
Fichiers des différents projets	7
Installer et utiliser les projets	8
CHAPITRE 1	
Jouons au taquin	9
Principe – Choix algorithmique	9
Recherche : A*	11
Le logiciel Python du taquin (8Puzzle) en pratique	14
Recherche A*	15
Apprentissage par renforcement : le Q-learning	16
Application au taquin	19
L'algorithme d'optimisation génétique	22
CHAPITRE 2	
Découvrir le plus court chemin	27
Principe – Choix algorithmique	28
Recherche du plus court chemin	28
Problème du voyageur de commerce	31
Le logiciel Python de la recherche du plus court chemin en pratique	33
CHAPITRE 3	
Jouons au sudoku	37
Principe – Choix algorithmique	38
Le logiciel Python du sudoku en pratique	42
Recherche best-first	42
Algorithme génétique	43

CHAPITRE 4

Jouons à Puissance 4	47
Principe – Choix algorithmique	47
Le logiciel Python de Puissance 4 en pratique	53
Min-Max	53
MCTS	56

CHAPITRE 5

Jouons au Snake	61
Principe – Choix algorithmique	61
Réseaux de neurones artificiels	64
Retour au Snake	68
Le logiciel Python du Snake en pratique	70
Recherche A*	70
Réseau de neurones et algorithme génétique	73

CHAPITRE 6

Jouons à Tetris	83
Principe – Choix algorithmique	84
L'apprentissage par renforcement pour le Tetris	85
Le logiciel Python du Tetris en pratique	89
Apprentissage par renforcement neuronal	89
Algorithme génétique	94

CHAPITRE 7

Comment reconnaître un spam.....	101
Principe – Choix algorithmique	102
Le logiciel Python du détecteur de spams en pratique	105
Classificateur naïf de Bayes	106

CHAPITRE 8

Découvrir les règles d'accès au crédit.....	109
Principe – Choix algorithmique	109
Démarrons par un petit récit personnel	109
Et l'apprentissage vint à la rescousse	111
Comment fonctionnent les arbres de décision ?	112
De l'arbre à la forêt	117
De l'éthique de l'apprentissage machine	117
L'arbre de décision en pratique	119
Construire un arbre de décision en utilisant sklearn	124

CHAPITRE 9

Aider à trier la presse ou les avis de clients 127

Principe – Choix algorithmique	127
Supervisé et non supervisé	127
L'algorithme LDA (Latent Dirichlet Allocation)	129
Word2Vec et Doc2Vec – Catégoriser le sens profond et non plus les mots	131
Le logiciel Python de la catégorisation de documents en pratique	132
LDA	132
Représenter les documents par un vecteur de « sens »	136

CHAPITRE 10

Comment distinguer un chien d'un chat 139

Principe – Choix algorithmique	140
Le logiciel Python de la reconnaissance chien/chat en pratique	147
Apprentissage profond	147

Conclusion : les deux IA 151**Index 159**

