

L'apprentissage profond avec Python

L'apprentissage profond avec Python

FRANÇOIS CHOLLET



machine
learning
.fr

L'apprentissage profond avec Python de François Chollet

Deep Learning with Python ISBN 978-1-61729-443-3 ©2017 François Chollet

Original English language edition published by Manning Publications



Copyright ©2017 by Manning Publications.

French edition copyright ©2020 by machinelearning.fr.

All rights reserved.

Collection **Les Essentiels de l'IA**

machinelearning.fr

47, rue de la Mésangerie

37540 Saint-Cyr sur Loire — France

Courriel : sales@machinelearning.fr



ISBN : 978-2-491674-00-7

©machinelearning.fr, 2020

Traduction de l'anglais : Jacqueline Isabelle Forien

Relecteur : Anthony Maranghi

Tous nos remerciements à nos lecteurs Franck Bardol et Jean-Michel Bernabotto

BRONX. Design graphique de la couverture : bronx.fr
Visuel © iStock/Vik_Y

Le Code de la propriété intellectuelle et artistique n'autorisant, aux termes des alinéas 2 et 3 de l'article L.122-5, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1er de l'article L. 122-4). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.

L'autorisation d'effectuer des reproductions par reprographie doit être obtenue auprès du Centre Français d'exploitation du droit de Copie (CFC – 20, rue des Grands- Augustins, 75006 Paris, Tél. : 01 44 07 47 70, Fax : 01 46 34 67 19).

Table des matières

Préface	i
Remerciements	iii
À propos de ce livre	iv
À propos de l'auteur	ix
À propos de la couverture des éditions Manning	x

Partie 1 Principes fondamentaux de l'apprentissage profond 1

1 Qu'est-ce que l'apprentissage profond ? 3

1.1 Intelligence artificielle, apprentissage automatique et apprentissage profond 5

- 1.1.1 Intelligence artificielle 5
- 1.1.2 Apprentissage automatique 6
- 1.1.3 Apprentissage des représentations à partir de données 8
- 1.1.4 Le « profond » de l'apprentissage profond 11
- 1.1.5 Comprendre le fonctionnement de l'apprentissage profond en trois graphiques 13
- 1.1.6 Ce que l'apprentissage profond est parvenu à réaliser jusqu'à présent 16
- 1.1.7 Ne vous fiez pas au buzz médiatique qui ne voit qu'à court terme 16
- 1.1.8 Les promesses de l'IA 18

1.2 Avant l'apprentissage profond : brève histoire de l'apprentissage automatique 20

- 1.2.1 Modélisation probabiliste 20
- 1.2.2 Premiers réseaux de neurones 21
- 1.2.3 Méthodes à noyau 21
- 1.2.4 Arbres de décision, forêts aléatoires et algorithmes de gradient boosting 23
- 1.2.5 Retour aux réseaux de neurones 25
- 1.2.6 Ce qui distingue l'apprentissage profond 26
- 1.2.7 Panorama de l'apprentissage automatique moderne 27

1.3 Pourquoi l'apprentissage profond ? Pourquoi maintenant ? 28

- 1.3.1 Le matériel (hardware) 28
- 1.3.2 Les données (data) 30
- 1.3.3 Les algorithmes 30
- 1.3.4 Une nouvelle vague d'investissements 31
- 1.3.5 La démocratisation de l'apprentissage profond 32
- 1.3.6 Quel avenir pour l'apprentissage profond ? 33

2 *Pour commencer : les notions mathématiques des réseaux de neurones* 35

2.1 Première approche d'un réseau de neurones 36

2.2 Représentations de données pour les réseaux de neurones 41

- 2.2.1 Scalaires (tenseurs 0D) 41
- 2.2.2 Vecteurs (tenseurs 1D) 41
- 2.2.3 Matrices (tenseurs 2D) 42
- 2.2.4 Tenseurs 3D et tenseurs de haute dimensionnalité 42
- 2.2.5 Principaux attributs d'un tenseur 43
- 2.2.6 Manipulation des tenseurs avec Numpy 44
- 2.2.7 Entraînement par lots de données 45
- 2.2.8 Exemples concrets de tenseurs de données 46
- 2.2.9 Données vectorielles 46
- 2.2.10 Données temporelles ou données séquentielles 47
- 2.2.11 Données d'images 48
- 2.2.12 Données vidéo 49

2.3 Outils des réseaux de neurones : les opérations de tenseurs 50

- 2.3.1 Opérations élément par élément (element-wise operations) 50
- 2.3.2 Broadcasting du langage Python 52
- 2.3.3 Produit scalaire de tenseurs 53
- 2.3.4 Modifier la forme d'un tenseur 57
- 2.3.5 Interprétation géométrique des opérations sur les tenseurs 58
- 2.3.6 Une interprétation géométrique de l'apprentissage profond 59

2.4 Le moteur des réseaux de neurones : l'optimisation à base de gradient 61

- 2.4.1 Qu'est-ce qu'une dérivée ? 62
- 2.4.2 Dérivée d'une opération de tenseurs : le gradient 63
- 2.4.3 Descente de gradient stochastique 64

2.4.4 Dérivation en chaîne de fonctions composées : l'algorithme de rétropropagation du gradient 68

2.5 Retour à notre premier exemple 70

2.6 Résumé du chapitre 72

3 *Mise en œuvre des réseaux de neurones sur des cas concrets* 73

3.1 Anatomie d'un réseau de neurones 75

3.1.1 Les couches : des briques de construction de réseaux 76

3.1.2 Les modèles : des réseaux de couches 77

3.1.3 Fonctions de perte et optimiseurs : les clés pour configurer un processus d'apprentissage 78

3.2 Introduction à la bibliothèque logicielle Keras 80

3.2.1 Bibliothèques Keras, TensorFlow, Theano et CNTK 81

3.2.2 Développement avec Keras : une brève introduction 82

3.3 Mise en route d'un poste de travail d'apprentissage profond 85

3.3.1 Les notebooks Jupyter : le meilleur moyen de mener des expériences d'apprentissage profond 85

3.3.2 Lancer l'exécution de Keras : deux options 86

3.3.3 Exécuter des travaux d'apprentissage profond dans l'informatique en nuage : avantages et inconvénients 86

3.3.4 Quel est le meilleur GPU pour l'apprentissage profond ? 87

3.4 Classement des critiques de films : un exemple de classification binaire 88

3.4.1 Le jeu de données IMDB 88

3.4.2 Préparation des données 90

3.4.3 Architecture du réseau 91

3.4.4 Validation de l'approche choisie 95

3.4.5 Utilisation d'un réseau pré-entraîné pour générer des prédictions sur de nouvelles données 99

3.4.6 Expériences plus avancées 100

3.4.7 Pour conclure cette section 100

3.5 Classification de fils de presse : un exemple de classification en plusieurs classes 102

- 3.5.1 Le jeu de données Reuters 102
- 3.5.2 Préparation des données 103
- 3.5.3 Architecture du réseau 104
- 3.5.4 Validation de l'approche choisie 106
- 3.5.5 Génération de prédictions sur de nouvelles données 109
- 3.5.6 Une approche alternative pour encoder les étiquettes et choisir la fonction de perte 109
- 3.5.7 De l'importance d'avoir des couches intermédiaires suffisamment larges 110
- 3.5.8 Expériences supplémentaires 111
- 3.5.9 Pour conclure cette section 111

3.6 Prédiction du prix de l'immobilier : un exemple de régression 112

- 3.6.1 Le jeu de données des prix de l'immobilier à Boston 112
- 3.6.2 Préparation des données 113
- 3.6.3 Architecture du réseau 114
- 3.6.4 Validation de l'approche à l'aide de la validation croisée à K-fold 115
- 3.6.5 Pour conclure cette section 120

3.7 Résumé du chapitre 121

4 Principes fondamentaux de l'apprentissage automatique 123

4.1 Quatre branches de l'apprentissage automatique 125

- 4.1.1 Apprentissage supervisé 125
- 4.1.2 Apprentissage non supervisé 126
- 4.1.3 Apprentissage auto-supervisé 126
- 4.1.4 Apprentissage par renforcement 127

4.2 Évaluation des modèles d'apprentissage automatique 130

- 4.2.1 Ensembles d'entraînement, de validation et de test 130
- 4.2.2 Points importants à garder à l'esprit 135

- 4.3 Prétraitement des données, extraction des caractéristiques et apprentissage des caractéristiques 137**
 - 4.3.1 Prétraitement des données pour les réseaux de neurones 137
 - 4.3.2 Extraction des caractéristiques 139

- 4.4 Sur-ajustement et sous-ajustement 142**
 - 4.4.1 Réduction de la taille du réseau 143
 - 4.4.2 Ajout d'une régularisation des poids 146
 - 4.4.3 Ajout de dropout 148

- 4.5 Le déroulement universel des tâches de l'apprentissage automatique 152**
 - 4.5.1 Définition du problème et constitution d'un jeu de données 152
 - 4.5.2 Choix d'une mesure de performance 153
 - 4.5.3 Choix d'un protocole d'évaluation 154
 - 4.5.4 Préparation des données 154
 - 4.5.5 Développement d'un modèle plus performant qu'un modèle de base 155
 - 4.5.6 Montée en puissance : développement d'un modèle sur-ajusté 157
 - 4.5.7 Régularisation du modèle et réglage des hyper-paramètres 157

- 4.6 Résumé du chapitre 159**

Partie 2 L'apprentissage profond en pratique 161

5 L'apprentissage profond pour la vision par ordinateur 163

- 5.1 Introduction aux réseaux de neurones convolutifs : les ConvNet 164**
 - 5.1.1 L'opération de convolution 167
 - 5.1.2 L'opération de max-pooling 174

5.2	Entraînement d'un ConvNet à partir de zéro sur un petit jeu de données	177
5.2.1	La pertinence de l'apprentissage profond pour des problèmes avec peu de données d'apprentissage	178
5.2.2	Téléchargement des données	178
5.2.3	Architecture du réseau	182
5.2.4	Prétraitement des données	184
5.2.5	Utilisation de l'augmentation de données	189
5.3	Utilisation d'un ConvNet pré-entraîné	196
5.3.1	Extraction de caractéristiques	197
5.3.2	Réglage de précision	208
5.3.3	Pour conclure ces deux dernières sections	216
5.4	Visualisation des représentations apprises par les ConvNet	217
5.4.1	Visualisation des cartes de caractéristiques des couches intermédiaires	217
5.4.2	Visualisation des filtres de ConvNet	226
5.4.3	Visualisation des cartes thermiques d'activation de classe (heatmaps of class activation)	233
5.5	Résumé du chapitre	240
6	<i>L'apprentissage profond pour les textes et les séquences</i>	241
6.1	Traitement des données textuelles	243
6.1.1	Représentation des mots et caractères avec des vecteurs one-hot	246
6.1.2	Utilisation des embeddings de mots	249
6.1.3	Synthèse : des textes bruts aux embeddings de mots	255
6.1.4	Pour conclure cette section	264
6.2	Comprendre les réseaux de neurones récurrents	265
6.2.1	Une couche récurrente dans Keras	268
6.2.2	Comprendre les couches LSTM et GRU	273
6.2.3	Un cas concret d'utilisation d'une couche LSTM dans Keras	277
6.2.4	Pour conclure cette section	279

6.3 Utilisation avancée des réseaux de neurones récurrents 280

- 6.3.1 Un problème de prédiction météorologique de la température 280
- 6.3.2 Préparation des données 283
- 6.3.3 Une base de référence de bon sens sans apprentissage automatique (non-machine-learning) 287
- 6.3.4 Une approche de base avec l'apprentissage automatique 288
- 6.3.5 Une première base de référence récurrente 290
- 6.3.6 Utilisation du dropout récurrent pour lutter contre le surajustement 292
- 6.3.7 Piles de couches récurrentes 293
- 6.3.8 Utilisation de RNN bidirectionnels 295
- 6.3.9 Autres stratégies pour améliorer les performances 300
- 6.3.10 Pour conclure cette section 300

6.4 Traitement des séquences avec des réseaux ConvNet 303

- 6.4.1 Comprendre la convolution 1D pour les séquences (sequence data) 303
- 6.4.2 Opération de pooling 1D pour les séquences (sequence data) 304
- 6.4.3 Implémentation d'un ConvNet 1D 305
- 6.4.4 Combiner des ConvNet et des RNN pour traiter de longues séquences 307
- 6.4.5 Pour conclure cette section 312

6.5 Résumé du chapitre 313***7 Les meilleures pratiques de l'apprentissage profond avancé 315*****7.1 Au-delà du modèle séquentiel : l'API Fonctionnelle de Keras 316**

- 7.1.1 Introduction à l'API Fonctionnelle 320
- 7.1.2 Modèles à plusieurs entrées 322
- 7.1.3 Modèles à plusieurs sorties 324
- 7.1.4 Graphes orientés acycliques de couches 328
- 7.1.5 Partage des poids d'une couche 333
- 7.1.6 Utilisation des modèles comme des couches 335
- 7.1.7 Pour conclure cette section 336

7.2 Inspection et surveillance des modèles à l'aide des callbacks de Keras et de TensorBoard 337

- 7.2.1 Utilisation des callbacks pour agir sur un modèle pendant son entraînement 337
- 7.2.2 Introduction à TensorBoard : l'outil de visualisation de TensorFlow 342
- 7.2.3 Pour conclure cette section 350

7.3 Comment tirer le meilleur parti de vos modèles 351

- 7.3.1 Architectures avancées 351
- 7.3.2 Optimisation des hyper-paramètres 355
- 7.3.3 Ensemblage de modèles 357
- 7.3.4 Pour conclure cette section 360

7.4 Résumé du chapitre 362

8 *L'apprentissage profond génératif* 363

8.1 Génération de texte avec un LSTM 366

- 8.1.1 Un bref historique des réseaux récurrents génératifs 366
- 8.1.2 Comment générer des données séquentielles 367
- 8.1.3 L'importance de la stratégie d'échantillonnage 368
- 8.1.4 Implémenter une génération de texte caractère par caractère avec un LSTM 370
- 8.1.5 Pour conclure cette section 376

8.2 L'algorithme DeepDream 377

- 8.2.1 Implémentation de DeepDream dans Keras 378
- 8.2.2 Pour conclure cette section 387

8.3 Transfert de style neuronal 388

- 8.3.1 La perte de contenu 389
- 8.3.2 La perte de style 390
- 8.3.3 Transfert de style neuronal dans Keras 390
- 8.3.4 Pour conclure cette section 399

8.4 Génération d'images avec des auto-encodeurs variationnels 400

- 8.4.1 Échantillonnage dans des espaces latents d'images 400
- 8.4.2 Vecteurs de concept pour l'édition d'images 402

- 8.4.3 Auto-encodeurs variationnels 402
- 8.4.4 Pour conclure cette section 410

8.5 Introduction aux réseaux génératifs antagonistes 412

- 8.5.1 Une implémentation schématique de GAN 414
- 8.5.2 Quelques astuces utiles 415
- 8.5.3 Le réseau génératif 416
- 8.5.4 Le réseau discriminant 417
- 8.5.5 Le réseau antagoniste 418
- 8.5.6 Comment entraîner un DCGAN 419
- 8.5.7 Pour conclure cette section 422

8.6 Résumé du chapitre 424

9 Conclusions 425

9.1 Révision des concepts clés 427

- 9.1.1 Différentes approches de l'IA 427
- 9.1.2 Ce qui fait la spécificité de l'apprentissage profond à l'intérieur du domaine de l'apprentissage automatique 427
- 9.1.3 Comment penser l'apprentissage profond 428
- 9.1.4 Technologies clés qui ont permis son développement 430
- 9.1.5 Le déroulement universel des tâches en apprentissage automatique 431
- 9.1.6 Architectures clés des réseaux 432
- 9.1.7 Le champ des applications possibles 437

9.2 Les limites de l'apprentissage profond 440

- 9.2.1 Le risque d'anthropomorphisation des modèles d'apprentissage automatique 441
- 9.2.2 Généralisation locale vs généralisation extrême 443
- 9.2.3 Pour conclure cette section 445

9.3 L'avenir de l'apprentissage profond 447

- 9.3.1 Des modèles qui ressembleront à des programmes 448
- 9.3.2 Au-delà de la rétropropagation et des couches différentiables 450
- 9.3.3 Apprentissage automatique automatisé 451
- 9.3.4 Apprentissage continu et réutilisation de sous-programmes modulaires 452
- 9.3.5 La vision à long terme 454

9.4 Rester à jour dans un domaine en évolution rapide 456

9.4.1 Pratiquer sur des problèmes concrets avec Kaggle 456

9.4.2 S'informer des derniers développements sur arXiv 456

9.4.3 Explorer l'écosystème Keras 457

9.5 Derniers mots 458

A Installer Keras et ses dépendances sur Ubuntu 459

B Exécution de notebooks Jupyter sur une instance GPU sur EC2 465

Index 473

Préface

Si vous avez choisi cet ouvrage, cela signifie probablement que vous êtes déjà conscient de la contribution extraordinaire que l'apprentissage profond a récemment apportée au domaine de l'intelligence artificielle (IA). En seulement cinq ans, nous sommes passés de performances quasi inutilisables dans les tâches de reconnaissance d'image et de transcription de la parole, à des performances surhumaines.

Les conséquences de ces progrès soudains s'étendent à presque tous les secteurs d'activités de l'industrie. Afin de commencer à déployer la technologie de l'apprentissage profond à tous les problèmes qu'elle pourrait résoudre, nous devons la rendre accessible au plus grand nombre de personnes possible, y compris aux non-experts — c'est-à-dire aux personnes qui ne sont ni des chercheurs, ni des étudiants de cycle supérieur. Pour que l'apprentissage profond atteigne son plein potentiel, nous devons radicalement le démocratiser.

Lorsque j'ai publié la première version de la bibliothèque logicielle d'apprentissage profond Keras en mars 2015, la démocratisation de l'IA n'était pas ma préoccupation. J'avais conduit des recherches en apprentissage automatique pendant plusieurs années, et j'avais développé Keras pour m'aider dans mes propres expérimentations. Mais, tout au long des années 2015 et 2016, des dizaines de milliers de nouvelles personnes se sont engagées dans le domaine de l'apprentissage profond ; beaucoup d'entre elles ont choisi Keras parce que c'était — et c'est toujours le cas — l'environnement le plus facile à utiliser pour débiter. À mesure que je voyais de nombreux nouveaux arrivants utiliser Keras de manière aussi puissante qu'inattendue, j'ai commencé à me préoccuper sérieusement de l'accessibilité et de la démocratisation de l'IA. Je me suis rendu compte que plus nous répandions ces technologies, plus elles devenaient utiles, plus elles avaient de la valeur. L'accessibilité est rapidement devenue un objectif explicite dans le développement de Keras et, en quelques années à peine, la communauté des développeurs de Keras a fait des progrès fantastiques sur ce front. Nous avons mis l'apprentissage profond entre les mains de dizaines de milliers de personnes qui, à leur tour, l'utilisent pour résoudre des problèmes importants dont nous ne soupçonnions même pas l'existence jusqu'à récemment.

Le livre que vous avez entre les mains est une nouvelle étape vers la mise à disposition de l'apprentissage profond au plus grand nombre. Keras a toujours eu besoin d'un cours d'accompagnement afin de couvrir simultanément les bases de l'apprentissage profond, les modes d'utilisation de Keras et les meilleures pratiques de l'apprentissage profond. Cet ouvrage est ce que j'ai accompli de plus élaboré pour atteindre cet objectif. Je l'ai écrit en mettant l'accent sur la nécessité de rendre

l'apprentissage profond, ses concepts sous-jacents et sa mise en œuvre aussi abordables que possible. Pour ce faire, il n'a pas été nécessaire d'abaisser le niveau d'exigence — je suis convaincu qu'il n'y a pas de réelles difficultés dans l'apprentissage profond. J'espère que vous trouverez ce livre profitable, et qu'il vous permettra de développer des applications intelligentes et de résoudre des problèmes qui vous tiennent à cœur.

Remerciements

J'aimerais remercier la communauté Keras d'avoir rendu possible la réalisation de ce livre. Keras compte aujourd'hui des centaines de contributeurs *Open Source* et plus de 200 000 utilisateurs. Leurs contributions et leurs commentaires ont fait de Keras ce qu'elle est aujourd'hui.

J'aimerais aussi remercier Google pour son soutien au projet Keras. Ce fut fantastique de voir Keras adoptée comme l'interface de programmation applicative (API) de haut niveau de TensorFlow. Une intégration harmonieuse entre Keras et TensorFlow profite grandement aussi bien aux utilisateurs de TensorFlow qu'à ceux de Keras. Cette intégration rend l'apprentissage profond accessible au plus grand nombre.

Je tiens à remercier les collaborateurs des éditions Manning qui ont rendu possible le livre de l'édition américaine : l'éditeur Marjan Bace et tous les membres des équipes de rédaction et de production, y compris Christina Taylor, Janet Vail, Tiffany Taylor, Katie Tennant, Dottie Marsico et beaucoup d'autres qui ont travaillé en coulisses.

Un grand merci aux évaluateurs techniques dirigés par Aleksandar Dragosavljević — Diego Acuña Rozas, Geoff Barto, David Blumenthal-Barby, Abel Brown, Clark Dorman, Clark Gaylord, Thomas Heiman, Wilson Mar, Sumit Pal, Vladimir Pasman, Gustavo Patino, Peter Rabinovitch, Alvin Raj, Claudio Rodriguez, Srdjan Santic, Richard Tobias, Martin Verzilli, William E. Wheeler, et Daniel Williams — et aux contributeurs du forum. Leurs contributions incluent la détection d'erreurs techniques, d'erreurs de terminologie et de fautes de frappe, ainsi que des suggestions de nouveaux thèmes. Chaque étape du processus de révision et chaque retour par le biais des thèmes abordés dans le forum ont façonné et modelé le manuscrit.

Sur le plan technique, des remerciements particuliers vont à Jerry Gaines qui a été le rédacteur technique du livre américain, et à Alex Ott et Richard Tobias qui en ont été les correcteurs techniques. Ce sont les meilleurs rédacteurs techniques que j'aurais pu espérer.

Enfin, j'aimerais exprimer ma gratitude à mon épouse Maria pour avoir été d'un grand soutien tout au long de la conception de Keras et de l'écriture de ce livre.

À propos de ce livre

Ce livre a été écrit pour tous ceux qui souhaitent explorer l'apprentissage profond à partir de zéro ou renforcer leur compréhension de l'apprentissage profond. Que vous soyez un ingénieur spécialisé en apprentissage automatique, un développeur de logiciels ou un étudiant, ces pages devraient vous être profitables.

Ce livre explore l'apprentissage profond de manière pratique et à partir d'applications concrètes. Il évite le recours à la notation mathématique, préférant expliquer les concepts quantitatifs à l'aide d'extraits de code. Il développe une intuition pratique des concepts de base de l'apprentissage automatique et de l'apprentissage profond.

Grâce à plus de 30 exemples de code, munis de commentaires détaillés, vous étudierez des recommandations pratiques et des explications simples de concepts de haut niveau. Vous saurez tout ce que vous avez besoin de savoir pour commencer à utiliser l'apprentissage profond, afin de résoudre des problèmes concrets.

Les exemples de code utilisent la bibliothèque logicielle Keras en Python, avec TensorFlow comme *backend engine*. Keras est l'une des bibliothèques logicielles d'apprentissage profond les plus populaires, avec la plus forte croissance. Elle est largement recommandée comme étant le meilleur outil pour commencer l'étude de l'apprentissage profond.

Après avoir lu ce livre, vous aurez une solide compréhension de ce qu'est l'apprentissage profond, des occasions où il est pertinent de l'utiliser et de ses limites. Vous serez familier avec le flux standard des tâches nécessaires pour aborder et résoudre des problèmes d'apprentissage automatique, et vous saurez comment résoudre les problèmes les plus couramment rencontrés. Vous pourrez utiliser Keras pour vous attaquer à des problèmes concrets allant de la vision par ordinateur au traitement automatique du langage naturel : classification d'images, prédiction de séries temporelles, analyse de sentiments, génération d'images et de textes, et bien d'autres applications.

Qui devrait lire ce livre ?

Ce livre est destiné aux personnes ayant de l'expérience en programmation Python, et qui veulent se lancer dans l'apprentissage automatique et dans l'apprentissage profond. Mais ce livre peut également profiter à de nombreux autres types de lecteurs :

- si vous êtes un spécialiste des données et connaissez bien l'apprentissage automatique, ce livre vous fournira une solide introduction pratique à l'apprentis-

sage profond, le sous-domaine de l'apprentissage automatique qui connaît la croissance la plus rapide et la plus forte ;

- si vous êtes un expert de l'apprentissage profond et souhaitez travailler avec la bibliothèque logicielle Keras, vous réaliserez que ce livre est le meilleur cours intensif disponible sur Keras ;
- si vous êtes un étudiant de cycle supérieur qui étudie l'apprentissage profond dans un cadre académique, ce livre sera un bon complément pratique à votre enseignement ; il vous aidera à développer votre intuition sur le comportement des réseaux de neurones profonds, et vous familiarisera avec les meilleures pratiques clés.

Même les personnes intéressées par la technique, mais qui ne programment pas régulièrement, trouveront ce livre utile en tant qu'introduction aussi bien aux concepts de base qu'aux concepts plus avancés de l'apprentissage profond.

Afin d'utiliser Keras, vous avez besoin d'une maîtrise raisonnable du langage de programmation Python. En complément, la connaissance de la bibliothèque Numpy sera utile, bien qu'elle ne soit pas strictement nécessaire. Vous n'avez pas besoin d'une expérience préalable de l'apprentissage automatique ni de l'apprentissage profond : ce livre couvre toutes les bases nécessaires. Vous n'avez pas besoin non plus d'une formation avancée en mathématiques — les mathématiques du niveau lycée devraient suffire pour que vous puissiez suivre.

Feuille de route

Ce livre est structuré en deux parties. Si vous n'avez aucune expérience préalable de l'apprentissage automatique, je vous recommande fortement de terminer la partie 1, avant d'aborder la partie 2. Nous commencerons par des exemples simples et, au fur et à mesure de la progression dans ce livre, nous nous rapprocherons de plus en plus des techniques de pointe.

La première partie est une introduction générale à l'apprentissage profond, précisant le contexte et les définitions, et expliquant toutes les notions nécessaires pour commencer l'étude de l'apprentissage automatique et des réseaux de neurones :

- Le chapitre 1 présente le contexte, essentiel à connaître, et les connaissances de base sur l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond.
- Le chapitre 2 introduit les concepts fondamentaux nécessaires pour aborder l'apprentissage profond : tenseurs, opérations sur les tenseurs, descente de gra-

dient et rétropropagation. Ce chapitre présente également le premier exemple d'un réseau de neurones en action.

- Le chapitre 3 contient tout ce que vous avez besoin de connaître pour commencer à utiliser les réseaux de neurones : une introduction à Keras qui sera notre infrastructure logicielle de référence pour l'apprentissage profond ; un guide d'installation pour configurer votre poste de travail ; et pour finir trois exemples de codes fondamentaux, munis d'explications détaillées. À la fin de ce chapitre, vous serez en mesure d'entraîner des réseaux de neurones simples pour effectuer des tâches de classification et de régression, et vous aurez également une idée précise de ce qui se déroule à l'arrière-plan pendant l'entraînement de ces réseaux.
- Le chapitre 4 étudie le flux canonique des tâches de l'apprentissage automatique. Vous découvrirez également les pièges courants et les remèdes pour les éviter.

La deuxième partie approfondit les applications pratiques de l'apprentissage profond qui sont adaptées à la vision par ordinateur et au traitement du langage naturel. Bon nombre des exemples présentés dans cette partie peuvent servir de modèles de référence pour résoudre des problèmes de mise en œuvre pratique de l'apprentissage profond :

- Le chapitre 5 examine une série d'exemples pratiques de vision par ordinateur, en mettant l'accent sur la classification d'images.
- Le chapitre 6 vous familiarise avec les techniques de traitement des données séquentielles telles que le texte et les séries temporelles.
- Le chapitre 7 présente des techniques avancées pour construire des modèles d'apprentissage profond à la pointe de la technologie.
- Le chapitre 8 décrit les modèles génératifs : ce sont des modèles d'apprentissage profond capables de générer des images et du texte, avec des résultats parfois étonnamment artistiques.
- Le chapitre 9 est consacré à la consolidation de ce que vous avez appris tout au long de ce livre. Il ouvre également des perspectives sur les limites de l'apprentissage profond et explore ses futurs développements.

Configuration logicielle et matérielle requise

Tous les programmes de ce livre utilisent la bibliothèque logicielle d'apprentissage profond Keras (<https://keras.io>) qui est disponible en *Open Source* et en libre téléchargement. Vous aurez besoin d'une machine tournant sous le système d'exploitation

UNIX. Il est aussi possible d'utiliser le système d'exploitation Windows, mais je ne le recommande pas. L'annexe A fournit un guide pas à pas pour une installation complète.

Je vous recommande également d'avoir à disposition sur votre machine un processeur graphique (GPU) récent de marque NVIDIA, comme un processeur graphique TITAN X par exemple. Cette configuration matérielle n'est pas obligatoire, mais elle améliorera grandement votre expérience en vous permettant d'exécuter beaucoup plus rapidement les exemples de code. La section 3.3 contient des informations complémentaires sur la configuration d'un poste de travail pour utiliser l'apprentissage profond.

Si vous n'avez pas accès à une station de travail locale avec un processeur graphique NVIDIA récent, vous pouvez utiliser, comme alternative, l'informatique en nuage (*cloud environment*). En particulier, vous pouvez utiliser des instances Google Cloud (par exemple, une instance n1-standard-8 avec un NVIDIA Tesla K80) ou des instances GPU sur Amazon Web Services (par exemple, une instance p2.xlarge). L'annexe B détaille le flux des tâches dans l'informatique en nuage. Elle présente l'exécution d'une instance AWS grâce à des *notebooks* Jupyter accessibles depuis votre navigateur.

Où trouver les exemples de code ?

Tous les exemples de code de ce livre peuvent être téléchargés sous la forme de *notebooks* Jupyter :

- sur le site Web du livre, www.manning.com/books/deep-learning-with-python ;
- et sur GitHub à <https://github.com/fchollet/deep-learning-with-python-notebooks>.

Forum du livre

Le livre *Deep Learning with Python* des éditions Manning (version américaine de cet ouvrage) inclut l'accès gratuit à un forum web privé géré par Manning Publications où vous pouvez publier des commentaires sur ce livre, poser des questions techniques, et recevoir de l'aide de l'auteur et des autres utilisateurs. Pour accéder au forum, rendez-vous sur <https://forums.manning.com/forums/deep-learning-with-python>. Vous pouvez également en apprendre davantage sur les forums de Manning et leurs règles de conduite à l'adresse <https://forums.manning.com/forums/about>.

L'engagement des éditions Manning envers ses lecteurs est de fournir un lieu où un dialogue constructif entre les lecteurs individuels, et entre les lecteurs et l'auteur

puisse avoir lieu. Il ne s'agit pas d'un engagement de participation de la part de l'auteur, dont la contribution au forum reste volontaire (et non rémunérée). Nous vous suggérons de le défier en lui posant quelques questions difficiles, de peur que son intérêt ne s'estompe ! Le forum et les archives des discussions précédentes seront accessibles sur le site web de l'éditeur tant que le livre sera imprimé.

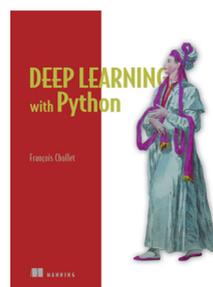
À propos de l'auteur

François Chollet est un ingénieur et chercheur en apprentissage profond chez Google à Mountain View, en Californie aux États-Unis. Il est le créateur de la bibliothèque d'apprentissage profond Keras, ainsi qu'un contributeur à la bibliothèque logicielle d'apprentissage automatique TensorFlow. Il mène également des recherches sur l'apprentissage profond, tout particulièrement sur la vision par ordinateur et l'application de l'apprentissage automatique au raisonnement formel. Ses articles sont publiés dans les plus grandes conférences de ce domaine, notamment la conférence *Computer Vision and Pattern Recognition (CVPR)*, la conférence *Neural Information Processing Systems (NeurIPS)*, la conférence *International Conference on Learning Representations (ICLR)* et bien d'autres événements de référence.



À propos de la couverture des éditions Manning

La figure sur la couverture du livre *Deep Learning with Python* (version américaine de ce livre) est intitulée « Habit d'une femme Perse en 1568 ». L'illustration est tirée de l'ouvrage de Thomas Jefferys appelé *A Collection of the Dresses of Different Nations, Ancient and Modern* (en quatre volumes), Londres, publié de 1757 à 1772. La page de garde indique qu'il s'agit de gravures en cuivre colorées à la main, rehaussées de gomme arabique.



Thomas Jefferys (1719-1771) était appelé le « Géographe du roi George III ». Cartographe anglais, il était le principal fournisseur de cartes de son époque. Il a gravé et imprimé des cartes pour le gouvernement et d'autres organismes officiels et a produit une vaste gamme de cartes et d'atlas commerciaux, surtout d'Amérique du Nord. Son travail de cartographe a développé son intérêt pour les coutumes vestimentaires locales des terres qu'il a arpentées et cartographiées, et qui sont brillamment illustrées dans sa collection. La fascination pour les terres lointaines et les voyages d'agrément étaient des phénomènes relativement nouveaux à la fin du XVIIIe siècle. Des collections comme celle-ci étaient populaires, présentant les habitants des autres pays, aussi bien aux touristes qu'aux voyageurs virtuels, lecteurs de l'ouvrage dans leur salon.

La diversité des illustrations dans les ouvrages de Jefferys témoigne de la singularité et de l'individualité des nations du monde d'il y a quelque 200 ans. Les codes vestimentaires ont changé depuis, et la diversité par région et par pays, si riche à l'époque, s'est estompée. Maintenant, il est souvent difficile de distinguer les habitants d'un continent à l'autre. Si nous essayons de voir cela de façon optimiste, nous avons peut-être échangé une diversité culturelle et visuelle contre une vie personnelle plus variée — ou une vie intellectuelle et technique plus variée et plus intéressante.

À une époque où il est difficile de distinguer un livre informatique d'un autre, Manning célèbre l'inventivité et l'initiative de l'industrie informatique avec des couvertures de livres basées sur la diversité abondante de la vie régionale d'il y a deux siècles, rendue à la vie par les illustrations de Jefferys.

Partie 1

Principes fondamentaux de l'apprentissage profond

Les chapitres 1 à 4 de ce livre vous aideront à acquérir une bonne compréhension des concepts fondamentaux de l'apprentissage profond, de ce qu'il peut accomplir et de son fonctionnement. Ils vous familiariseront aussi avec le flux canonique des tâches nécessaires à la résolution de problèmes avec l'apprentissage profond. Si vous n'avez pas déjà une grande expérience de l'apprentissage profond, vous devez absolument commencer par lire la partie 1 dans son intégralité, avant de passer aux applications pratiques de la partie 2.

Qu'est-ce que l'apprentissage profond ?

Dans ce chapitre, vous découvrirez :

- les définitions des concepts fondamentaux ;
- la chronologie du développement de l'apprentissage automatique ;
- les facteurs clés de la popularité croissante de l'apprentissage profond et ses évolutions futures.

Au cours de ces dernières années, l'intelligence artificielle (IA) a fait l'objet d'un buzz médiatique intense. L'apprentissage automatique (*machine learning*), l'apprentissage profond (*deep learning*) et l'intelligence artificielle (*artificial intelligence*) apparaissent dans d'innombrables articles, souvent en dehors de publications axées sur la technologie. On nous promet un avenir d'agents conversationnels intelligents (*chat-bots*), de voitures autonomes et d'assistants virtuels — un avenir parfois peint sous un jour sombre, et d'autres fois utopique, où les emplois humains seront rares, et où la plupart des activités économiques seront gérées par des robots ou par des agents IA (*AI agents*). Pour un utilisateur de l'apprentissage automatique, d'aujourd'hui ou de demain, il est important de pouvoir reconnaître le signal au milieu du bruit, afin de pouvoir distinguer les développements qui transforment le monde, des communiqués de presse à sensation. Notre futur est en jeu, et c'est un futur dans lequel vous avez un rôle actif à jouer : après avoir lu ce livre, vous ferez partie de ceux qui développent des agents IA (*AI agents*). Abordons donc ces questions : qu'est parvenu à réaliser l'apprentissage profond jusqu'à présent ? Quel est son impact réel ? Vers quoi nous dirigeons-nous ? Faut-il en croire le buzz médiatique ?

Ce chapitre fournit les éléments essentiels pour comprendre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond.

1.1 Intelligence artificielle, apprentissage automatique et apprentissage profond

Premièrement, nous devons définir clairement ce dont nous parlons lorsqu'il est question d'IA. Que sont l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond (voir figure 1.1) ? Quels liens existent entre eux ?

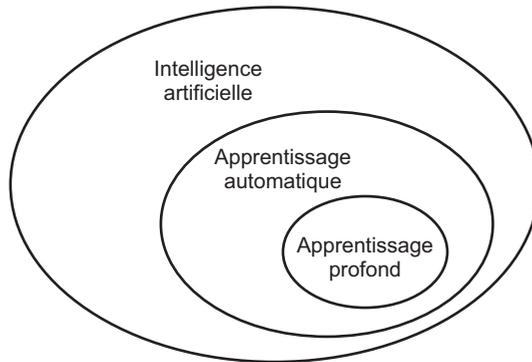


Figure 1.1 Les relations entre l'intelligence artificielle, l'apprentissage automatique et l'apprentissage profond

1.1.1 Intelligence artificielle

L'intelligence artificielle est née dans les années 50, quand une poignée de pionniers de l'informatique naissante ont commencé à se demander si les ordinateurs pouvaient être conçus pour « penser » — une question dont nous continuons aujourd'hui d'explorer les ramifications. Une définition précise de ce domaine serait la suivante : **c'est l'effort d'automatisation des tâches intellectuelles normalement effectuées par des humains**. L'intelligence artificielle est donc un domaine général qui englobe l'apprentissage automatique et l'apprentissage profond, mais qui comprend également de nombreuses autres approches qui n'impliquent aucun apprentissage. Par exemple, les premiers programmes d'échecs ne mettaient en oeuvre que des règles codées en dur (*hardcoded*) élaborées par des programmeurs, et qui ne pouvaient pas être qualifiées d'apprentissage automatique. Pendant très longtemps, de nombreux experts ont cru qu'une intelligence artificielle équivalente à celle de l'homme pouvait être atteinte en faisant en sorte que les programmeurs fabriquent un ensemble suffisamment large de règles explicites pour manipuler les connaissances. Cette approche est connue sous le nom d'**intelligence artificielle symbolique** (*symbolic AI*), et ce fut le paradigme

dominant de l'IA des années 1950 à la fin des années 1980. Elle a atteint son apogée pendant le boom **des systèmes experts** (*expert systems*) dans les années 1980.

Bien que l'IA symbolique se soit révélée apte à résoudre des problèmes logiques bien définis, tels que jouer aux échecs, elle s'est avérée incapable de définir des règles explicites pour résoudre des problèmes plus complexes et flous, tels que la classification d'images, la reconnaissance de la parole et la traduction linguistique. Une nouvelle approche est apparue, et elle a pris la place de l'intelligence artificielle symbolique : c'est **l'apprentissage automatique** (*machine learning*).

1.1.2 Apprentissage automatique

Dans l'Angleterre victorienne, Lady Ada Lovelace était une amie et une collaboratrice de Charles Babbage, l'inventeur de **la machine analytique** (*Analytical Engine*) : le premier ordinateur mécanique polyvalent connu. Bien que visionnaire et très en avance sur son temps, la machine analytique n'était pas conçue comme un ordinateur polyvalent (*general purpose computer*) lorsqu'elle a été pensée dans les années 1830 et 1840, car le concept de calcul polyvalent n'avait pas encore été inventé. C'était simplement un moyen d'utiliser des opérations mécaniques pour automatiser certains calculs du domaine de l'analyse mathématique — d'où le nom de machine analytique. En 1843, Ada Lovelace a ainsi commenté l'invention : « le moteur analytique n'a aucune prétention à être à l'origine de quoi que ce soit. Il peut faire toutes les tâches dont nous savons comment lui ordonner de les effectuer [...]. Son domaine de compétence est de nous aider à rendre disponible ce que nous connaissons déjà. »

Le pionnier de l'IA, Alan Turing, a cité cette remarque plus tard comme « l'objection de Lady Lovelace » dans son article phare de 1950 appelé « Computing Machinery and Intelligence »¹ qui présentait **le test de Turing** ainsi que les concepts clés qui façonneront plus tard l'IA. Turing citait Ada Lovelace en se demandant si les ordinateurs polyvalents pourraient être un jour capables d'apprendre et de faire preuve de créativité, et il en vint à la conclusion qu'ils en seraient capables.

L'apprentissage automatique découle de cette question : un ordinateur pourrait-il aller au-delà des « tâches dont nous savons comment lui ordonner de les effectuer », et apprendre par lui-même comment effectuer une tâche spécifique ? Un ordinateur pourrait-il nous surprendre ? À la place de programmeurs élaborant à la main des règles de traitement de données, un ordinateur pourrait-il apprendre automatiquement ces règles par l'exposition aux données ?

Cette question ouvre la porte à un nouveau paradigme de programmation. En programmation classique, le paradigme de l'IA symbolique, l'homme saisit des règles

¹ A. M. Turing, "Computing Machinery and Intelligence," *Mind* 59, no. 236 (1950) : 433-460.

(un programme) et des données à traiter conformément à ces règles, et il en découle des réponses en sortie (voir figure 1.2). Avec l'apprentissage automatique, les humains entrent des données, ainsi que les réponses attendues à partir de ces données, et ils obtiennent des règles en sortie. Ces règles peuvent ensuite être appliquées à de nouvelles données pour produire des réponses originales.

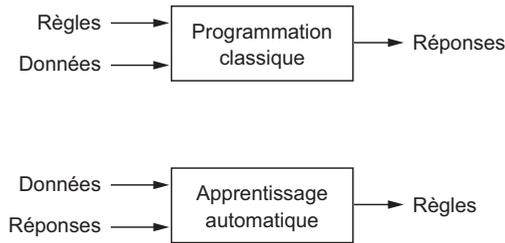


Figure 1.2 *L'apprentissage profond : un nouveau paradigme de programmation*

Un système d'apprentissage automatique est **entraîné** plutôt qu'explicitement programmé. De nombreux exemples pertinents pour une tâche lui sont présentés. Puis il trouve dans ces exemples une structure statistique qui lui permet de élaborer des règles pour l'automatisation de la tâche. Par exemple, si vous souhaitez automatiser l'étiquetage de vos photos de vacances, vous pouvez présenter à un système d'apprentissage automatique de nombreux exemples d'images déjà étiquetées par des humains, et le système apprendra des règles statistiques pour associer, à des images spécifiques, des étiquettes spécifiques.

Bien que l'apprentissage automatique n'ait commencé à prospérer que dans les années 1990, il est rapidement devenu le sous-domaine de l'IA le plus populaire et le plus performant. Cette tendance est alimentée par la disponibilité de matériels informatiques plus rapides et de plus grands ensembles de données. L'apprentissage automatique est étroitement lié aux statistiques mathématiques, mais il diffère des statistiques sur plusieurs points importants. À la différence des statistiques, l'apprentissage automatique traite généralement de vastes et complexes ensembles de données (par exemple un ensemble de données de millions d'images, chacune comprenant des dizaines de milliers de pixels) pour lesquels une analyse statistique classique telle qu'une analyse bayésienne serait impossible à mettre en œuvre. En conséquence, l'apprentissage automatique, et en particulier l'apprentissage profond, présente relativement peu de théorie mathématique — peut-être trop peu — et est axé sur l'ingénierie. C'est une discipline pratique dans laquelle les idées sont plus souvent prouvées empiriquement que théoriquement.

1.1.3 Apprentissage des représentations à partir de données

Pour définir l'**apprentissage profond** (*deep learning*) et comprendre la différence entre l'apprentissage profond et les autres approches d'apprentissage automatique, nous devons d'abord avoir une idée du **fonctionnement** des algorithmes d'apprentissage automatique. Nous venons juste d'énoncer que l'apprentissage automatique découvre des règles permettant d'exécuter une tâche de traitement de données, lorsque lui sont fournis des exemples de résultats attendus. Pour faire de l'apprentissage automatique, nous avons donc besoin de trois choses :

- **des points de données d'entrée** (*input data points*) — par exemple, si la tâche est la reconnaissance vocale, ces points de données peuvent être des fichiers audio de personnes qui parlent ; si la tâche est l'étiquetage d'images, ces points de données peuvent être des images ;
- **des exemples de sortie attendue** (*expected output*) — dans une tâche de reconnaissance vocale, il peut s'agir de transcriptions de fichiers sonores générés par l'homme ; dans une tâche d'étiquetage d'images, les sorties attendues peuvent être des étiquettes telles que « chien », « chat », etc. ;
- **un moyen de mesurer la performance de l'algorithme** — c'est un élément nécessaire pour déterminer la distance, au sens mathématique, entre la sortie effective de l'algorithme et la sortie attendue ; la mesure est utilisée comme un signal de retour (*feedback*) pour ajuster le fonctionnement de l'algorithme ; cette étape d'ajustement est ce que nous appelons **l'apprentissage**.

Un modèle d'apprentissage automatique transforme ses données d'entrée en sorties qui ont un sens, c'est un processus qui est « appris » à partir de l'exposition à des exemples connus d'entrées et de sorties. Par conséquent, le problème central de l'apprentissage automatique et de l'apprentissage profond est de **transformer de manière utile les données** : en d'autres termes, d'apprendre des **représentations** utiles des données d'entrée disponibles — des représentations qui nous rapprochent du résultat attendu. Avant d'aller plus loin : qu'est-ce qu'une représentation ? Fondamentalement, c'est une façon différente de considérer les données — de **représenter** ou **d'encoder** les données. Par exemple, une image en couleur peut être codée au format RGB (*Red Green Blue*) ou au format HSV (*Hue Saturation Value*) : il s'agit de deux représentations différentes des mêmes données. Certaines tâches qui peuvent être difficiles avec une représentation peuvent devenir faciles avec une autre. Par exemple, la tâche « sélectionner tous les pixels rouges de l'image » est plus simple au format RGB, tandis que la tâche « rendre l'image moins saturée » est plus simple au format HSV. Les modèles d'apprentissage automatique ont pour but de trouver des représentations appropriées pour leurs données d'entrée — des transformations de données qui

les rendent plus adaptées à la tâche à accomplir, telle que par exemple une tâche de classification.

Rendons cela plus concret. Considérons un axe des abscisses x , un axe des ordonnées y et certains points représentés par leurs coordonnées dans le système (x, y) , comme le montre la figure 1.3.

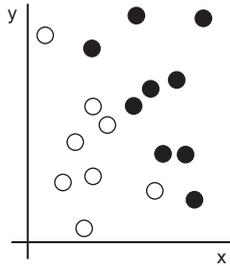


Figure 1.3 *Quelques exemples de données*

Comme vous pouvez le constater, nous avons quelques points blancs et quelques points noirs. Supposons que nous voulions développer un algorithme capable de prendre les coordonnées (x, y) d'un point et de déduire si ce point est susceptible d'être noir ou blanc. Dans ce cas :

- les entrées sont les coordonnées de nos points ;
- les sorties attendues sont les couleurs de nos points ;
- un moyen de mesurer si notre algorithme fait du bon travail pourrait être, par exemple, le pourcentage de points correctement classés.

Ce dont nous avons besoin ici, c'est d'une nouvelle représentation de nos données qui sépare correctement les points blancs des points noirs. Une transformation que nous pourrions utiliser, parmi beaucoup d'autres, serait un changement de coordonnées, illustré par la figure 1.4.

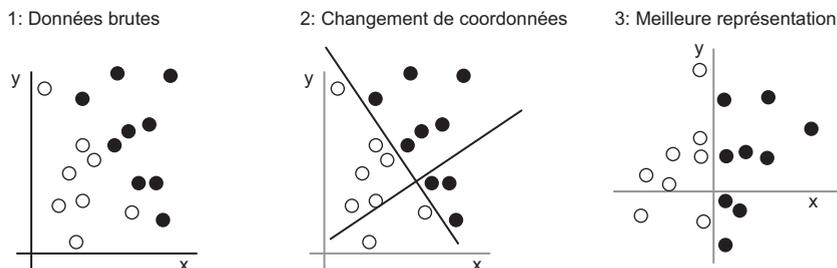


Figure 1.4 *Changement de coordonnées*

Dans ce nouveau système de coordonnées, les coordonnées de nos points peuvent être considérées comme une nouvelle représentation des données. Et c'est une bonne représentation ! Avec cette représentation, le problème de classification noir / blanc peut être exprimé par une règle simple : « les points noirs sont ceux tels que $x > 0$ » ou « les points blancs sont ceux tels que $x < 0$ ». Cette nouvelle représentation résout de fait le problème de classification.

Dans ce cas, nous avons défini manuellement le changement des coordonnées. Mais si, au lieu de cette méthode, nous essayions systématiquement de rechercher différents changements de coordonnées, et si nous utilisons comme retour le pourcentage des points correctement classés, nous réaliserions alors la même tâche que celle de l'apprentissage automatique. **L'apprentissage**, dans le contexte de l'apprentissage automatique, décrit un processus de recherche automatique des meilleures représentations.

Tous les algorithmes d'apprentissage automatique consistent à trouver automatiquement de telles transformations qui changent les données en des représentations plus utiles pour une tâche donnée. Ces opérations peuvent être des changements de coordonnées, comme vous venez de le voir, ou des projections linéaires (qui peuvent détruire des informations), des translations, des opérations non linéaires (telles que « sélectionner tous les points tels que $x > 0$ »), et ainsi de suite. Les algorithmes d'apprentissage automatique ne font généralement pas preuve de créativité pour trouver ces transformations. Ils se contentent d'effectuer une recherche dans un ensemble prédéfini d'opérations qui est appelé **l'espace d'hypothèses** (*hypothesis space*).

Techniquement, voici ce qu'est l'apprentissage automatique : c'est la recherche de représentations utiles de certaines données d'entrée, dans un espace des possibilités prédéfini, en s'appuyant sur un signal de retour. Cette idée simple permet de résoudre un très large éventail de tâches intellectuelles, de la reconnaissance automatique de la parole à la conduite automobile autonome.

Maintenant que vous comprenez ce que nous entendons par **apprentissage**, examinons ce qui fait la spécificité de **l'apprentissage profond**.

1.1.4 Le « profond » de l'apprentissage profond

L'apprentissage profond est un sous-domaine spécifique de l'apprentissage automatique : c'est une nouvelle approche de l'apprentissage des représentations à partir des données qui met l'accent sur l'apprentissage de **couches** (*layers*) successives de représentations qui sont de plus en plus significatives. L'adjectif « **profond** » de **l'apprentissage profond** ne fait pas référence à une forme de compréhension plus approfondie réalisée par l'approche mise en œuvre ; il représente plutôt l'idée de couches successives de représentations. Le nombre de couches qui contribuent à un modèle de données s'appelle **la profondeur** (*depth*) du modèle. D'autres noms appropriés pour l'apprentissage profond auraient pu être **l'apprentissage des représentations en couches** (*layered representations learning*) et **l'apprentissage des représentations hiérarchiques** (*hierarchical representations learning*). L'apprentissage profond moderne comprend souvent des dizaines, voire des centaines, de couches successives de représentations — celles-ci apprennent toutes automatiquement à partir de l'exposition aux données d'apprentissage. Cependant, d'autres approches de l'apprentissage automatique tendent à privilégier l'apprentissage d'une ou deux couches de représentations de données ; c'est pourquoi elles sont parfois nommées **apprentissages non profonds** (*shallow learning*).

En apprentissage profond, ces représentations en couches sont (presque toujours) apprises au moyen de modèles appelés **réseaux de neurones** (*neural networks*), structurés en couches littéralement superposées les unes sur les autres. L'expression **réseau de neurones** est une référence à la neurobiologie mais, bien que certains concepts fondamentaux de l'apprentissage profond aient été développés en partie en s'inspirant de notre compréhension du cerveau, les modèles d'apprentissage profond ne sont pas des modèles du cerveau. Rien n'indique que le cerveau mette en œuvre des mécanismes d'apprentissage semblables à ceux utilisés dans les modèles modernes d'apprentissage profond. Il se peut que vous rencontriez des articles de vulgarisation scientifique proclamant que l'apprentissage profond fonctionne comme le cerveau ou a été conçu à l'image du cerveau, mais ce n'est pas le cas. Il serait déroutant et contre-productif pour les nouveaux venus dans le domaine de l'apprentissage profond de le penser comme étant d'une quelconque manière lié à la neurobiologie. Il n'est pas nécessaire d'avoir ce voile de mysticisme et de mystère associé à l'idée qu'il est « juste comme notre esprit », et vous pouvez aussi oublier tout ce que vous avez pu lire sur les liens hypothétiques entre l'apprentissage profond et la biologie. Pour nos besoins ici, l'apprentissage profond est un cadre mathématique pour l'apprentissage des représentations à partir de données.

À quoi ressemblent les représentations apprises par un algorithme d'apprentissage profond ? Examinons maintenant comment un réseau profond de plusieurs couches (voir la figure 1.5) transforme l'image d'un chiffre afin de le reconnaître.

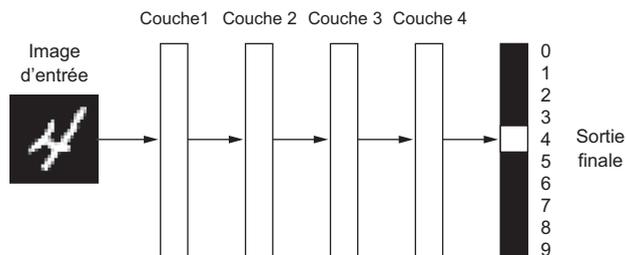


Figure 1.5 Un réseau de neurones profond pour la classification des chiffres

Comme vous pouvez le voir sur la figure 1.6, le réseau transforme l'image numérique en représentations de plus en plus différentes de l'image d'origine et de plus en plus riches d'informations sur le résultat final. Vous pouvez considérer un réseau profond comme une opération de distillation d'informations en plusieurs étapes, où l'information passe par des filtres successifs et ressort de plus en plus **épurée** (c'est-à-dire de plus en plus utile pour une tâche donnée).

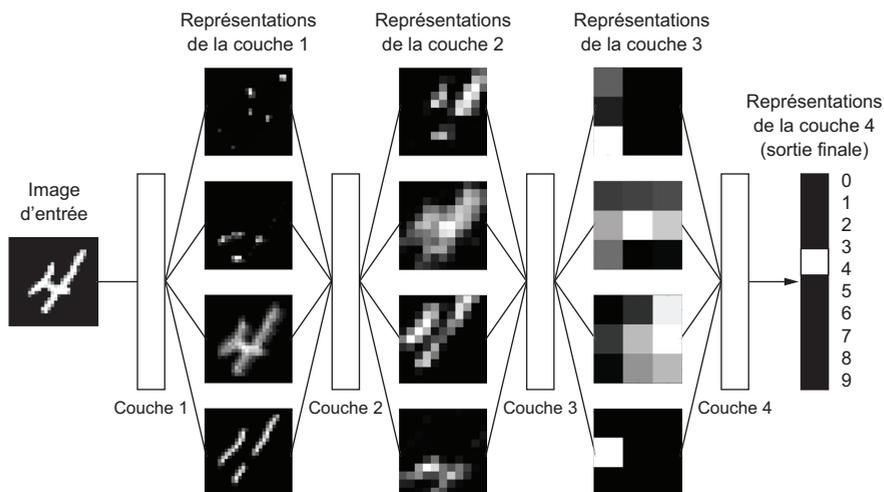


Figure 1.6 Des représentations profondes apprises par un modèle de classification de chiffres

Voilà donc ce qu'est l'apprentissage profond au plan technique : une méthode à plusieurs étapes pour apprendre des représentations de données. C'est une idée simple, mais il se trouve que des mécanismes très simples, suffisamment bien dimensionnés, peuvent finir par ressembler à de la magie.

1.1.5 Comprendre le fonctionnement de l'apprentissage profond en trois graphiques

À ce stade, vous savez que l'apprentissage automatique consiste à mettre en correspondance des entrées (telles que des images) avec des étiquettes cibles (*targets*) (telles que l'étiquette « chat »), ce qui s'obtient en observant de nombreux exemples d'entrées et de cibles. Vous savez également que les réseaux de neurones profonds effectuent ce mappage entrées-cibles (*input-to-target mapping*) par une série de transformations de données simples (couches) et que ces transformations de données sont apprises grâce à l'exposition à des exemples. Voyons maintenant comment cet apprentissage se déroule concrètement.

La spécification de ce que fait une couche sur ses données d'entrée est stockée dans **les poids** (*weights*) de la couche qui sont en substance un ensemble de nombres. En termes techniques, nous dirions que la transformation mise en œuvre par une couche est **paramétrée** par ses poids (voir figure 1.7). (Les poids sont parfois appelés **les paramètres** de la couche.) Dans ce contexte, **apprendre** signifie rechercher un ensemble de valeurs pour les poids de toutes les couches dans un réseau, de sorte que le réseau met correctement en correspondance les exemples d'entrées avec les étiquettes cibles qui leur sont associées. Mais voilà : un réseau de neurones profonds peut contenir des dizaines de millions de paramètres. Trouver la valeur correcte pour chacun d'entre eux peut sembler un défi de taille, d'autant plus que la modification de la valeur d'un paramètre affectera le comportement de tous les autres !

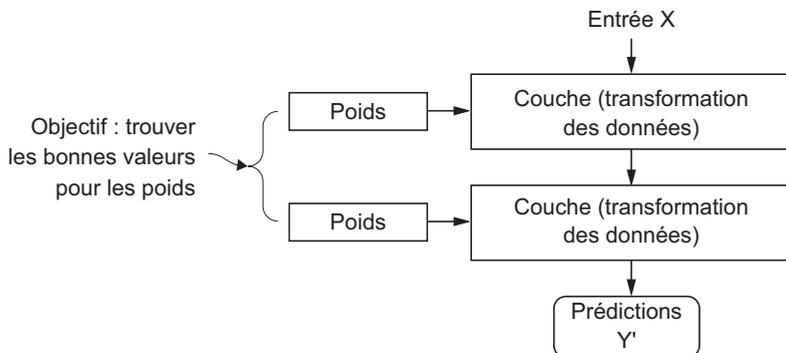


Figure 1.7 Un réseau de neurones est paramétré par ses poids.

Pour contrôler un événement, il faut d'abord être capable de l'observer. Pour contrôler la sortie d'un réseau de neurones, vous devez pouvoir mesurer dans quelle mesure cette sortie est différente de celle que vous attendiez. C'est le travail de **la fonction de perte** (*loss function*) du réseau, parfois appelée **la fonction objectif** (*objective function*). La fonction de perte prend les prédictions du réseau et la cible réellement attendue (ce que vous vouliez que le réseau vous retourne) et calcule un score de distance, capturant ainsi les performances du réseau pour cet exemple spécifique (voir la figure 1.8).

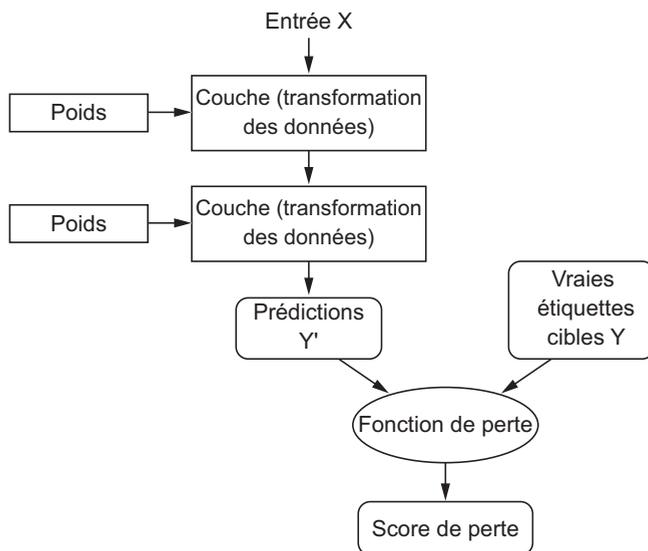


Figure 1.8 Une fonction de perte mesure la qualité des données de sortie du réseau.

L'astuce fondamentale de l'apprentissage profond est d'utiliser ce score comme un signal de retour (*feedback*) pour ajuster légèrement la valeur des poids dans un sens qui réduira le score de perte relatif à l'exemple actuel (voir la figure 1.9). Cet ajustement est le travail de l'**optimiseur** (*optimizer*) qui met en œuvre ce que l'on appelle l'algorithme de **rétropropagation** (*backpropagation*) : c'est l'algorithme central de l'apprentissage profond. Le chapitre suivant explique plus en détail comment fonctionne la rétropropagation.

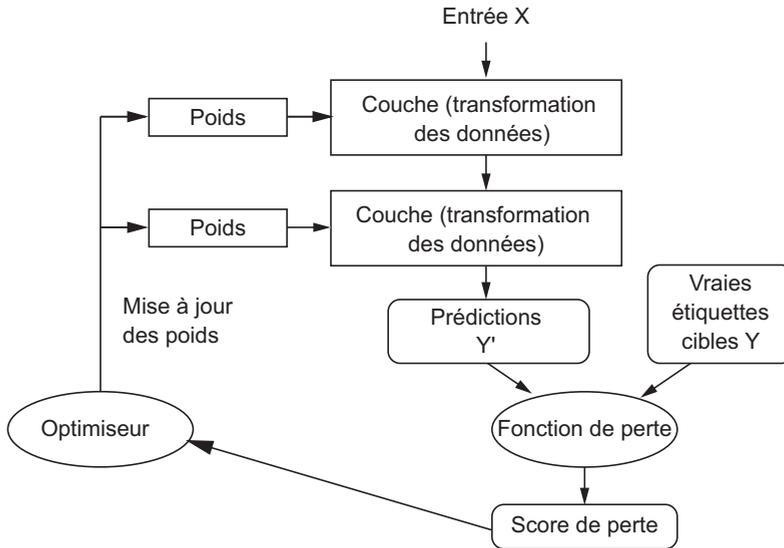


Figure 1.9 Le score de perte sert de signal de retour pour ajuster les poids.

Initialement, des valeurs aléatoires sont attribuées aux poids du réseau, de sorte que le réseau implémente simplement une série de transformations aléatoires. Naturellement, ses données de sortie sont loin de ce qu'elles devraient être idéalement, et le score de perte est donc très élevé. Mais, au fil des exemples traités, les poids sont légèrement ajustés dans la bonne direction, et le score de perte diminue. Il s'agit de **la boucle d'apprentissage** (*training loop*) qui, répétée un nombre suffisant de fois (généralement des dizaines d'itérations sur des milliers d'exemples), produit des valeurs de poids qui minimisent la fonction de perte. Un réseau avec une perte minimale est un réseau dont les sorties sont aussi proches que possible des cibles : c'est-à-dire un réseau entraîné. À nouveau, c'est un mécanisme simple qui, une fois démultiplié, finit par ressembler à de la magie.