

Machine Learning Refined

With its intuitive yet rigorous approach to machine learning, this text provides students with the fundamental knowledge and practical tools needed to conduct research and build data-driven products. The authors prioritize geometric intuition and algorithmic thinking, and include detail on all the essential mathematical prerequisites, to offer a fresh and accessible way to learn. Practical applications are emphasized, with examples from disciplines including computer vision, natural language processing, economics, neuroscience, recommender systems, physics, and biology. Over 300 color illustrations are included and have been meticulously designed to enable an intuitive grasp of technical concepts, and over 100 in-depth coding exercises (in Python) provide a real understanding of crucial machine learning algorithms. A suite of online resources including sample code, data sets, interactive lecture slides, and a solutions manual are provided online, making this an ideal text both for graduate courses on machine learning and for individual reference and self-study.

Jeremy Watt received his PhD in Electrical Engineering from Northwestern University, and is now a machine learning consultant and educator. He teaches machine learning, deep learning, mathematical optimization, and reinforcement learning at Northwestern University.

Reza Borhani received his PhD in Electrical Engineering from Northwestern University, and is now a machine learning consultant and educator. He teaches a variety of courses in machine learning and deep learning at Northwestern University.

Aggelos K. Katsaggelos is the Joseph Cummings Professor at Northwestern University, where he heads the Image and Video Processing Laboratory. He is a Fellow of IEEE, SPIE, EURASIP, and OSA and the recipient of the IEEE Third Millennium Medal (2000).

Machine Learning Refined

Foundations, Algorithms, and Applications

JEREMY WATT

Northwestern University, Illinois

REZA BORHANI

Northwestern University, Illinois

AGGELOS K. KATSAGGELOS

Northwestern University, Illinois



CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

One Liberty Plaza, 20th Floor, New York, NY 10006, USA

477 Williamstown Road, Port Melbourne, VIC 3207, Australia

314–321, 3rd Floor, Plot 3, Splendor Forum, Jasola District Centre, New Delhi – 110025, India

79 Anson Road, #06–04/06, Singapore 079906

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning, and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9781108480727

DOI: 10.1017/9781108690935

© Cambridge University Press 2020

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2020

Printed and bound in Great Britain by Clays Ltd, Elcograf S.p.A.

A catalogue record for this publication is available from the British Library.

ISBN 978-1-108-48072-7 Hardback

Additional resources for this publication at www.cambridge.org/watt2

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication and does not guarantee that any content on such websites is, or will remain, accurate or appropriate.

To our families:

Deb, Robert, and Terri

Soheila, Ali, and Maryam

Ειρηνη, Ζωη, Σοφια, and Ειρηνη

Cambridge University Press
978-1-108-48072-7 — Machine Learning Refined
2nd Edition
Frontmatter
[More Information](#)

Contents

	<i>Preface</i>	page xii
	<i>Acknowledgements</i>	xxii
1	Introduction to Machine Learning	1
	1.1 Introduction	1
	1.2 Distinguishing Cats from Dogs: a Machine Learning Approach	1
	1.3 The Basic Taxonomy of Machine Learning Problems	6
	1.4 Mathematical Optimization	16
	1.5 Conclusion	18
	Part I Mathematical Optimization	19
2	Zero-Order Optimization Techniques	21
	2.1 Introduction	21
	2.2 The Zero-Order Optimality Condition	23
	2.3 Global Optimization Methods	24
	2.4 Local Optimization Methods	27
	2.5 Random Search	31
	2.6 Coordinate Search and Descent	39
	2.7 Conclusion	40
	2.8 Exercises	42
3	First-Order Optimization Techniques	45
	3.1 Introduction	45
	3.2 The First-Order Optimality Condition	45
	3.3 The Geometry of First-Order Taylor Series	52
	3.4 Computing Gradients Efficiently	55
	3.5 Gradient Descent	56
	3.6 Two Natural Weaknesses of Gradient Descent	65
	3.7 Conclusion	71
	3.8 Exercises	71
4	Second-Order Optimization Techniques	75
	4.1 The Second-Order Optimality Condition	75

4.2	The Geometry of Second-Order Taylor Series	78
4.3	Newton's Method	81
4.4	Two Natural Weaknesses of Newton's Method	90
4.5	Conclusion	91
4.6	Exercises	92
Part II	Linear Learning	97
5	Linear Regression	99
5.1	Introduction	99
5.2	Least Squares Linear Regression	99
5.3	Least Absolute Deviations	108
5.4	Regression Quality Metrics	111
5.5	Weighted Regression	113
5.6	Multi-Output Regression	116
5.7	Conclusion	120
5.8	Exercises	121
5.9	Endnotes	124
6	Linear Two-Class Classification	125
6.1	Introduction	125
6.2	Logistic Regression and the Cross Entropy Cost	125
6.3	Logistic Regression and the Softmax Cost	135
6.4	The Perceptron	140
6.5	Support Vector Machines	150
6.6	Which Approach Produces the Best Results?	157
6.7	The Categorical Cross Entropy Cost	158
6.8	Classification Quality Metrics	160
6.9	Weighted Two-Class Classification	167
6.10	Conclusion	170
6.11	Exercises	171
7	Linear Multi-Class Classification	174
7.1	Introduction	174
7.2	One-versus-All Multi-Class Classification	174
7.3	Multi-Class Classification and the Perceptron	184
7.4	Which Approach Produces the Best Results?	192
7.5	The Categorical Cross Entropy Cost Function	193
7.6	Classification Quality Metrics	198
7.7	Weighted Multi-Class Classification	202
7.8	Stochastic and Mini-Batch Learning	203
7.9	Conclusion	205
7.10	Exercises	205

8	Linear Unsupervised Learning	208
	8.1 Introduction	208
	8.2 Fixed Spanning Sets, Orthonormality, and Projections	208
	8.3 The Linear Autoencoder and Principal Component Analysis	213
	8.4 Recommender Systems	219
	8.5 K-Means Clustering	221
	8.6 General Matrix Factorization Techniques	227
	8.7 Conclusion	230
	8.8 Exercises	231
	8.9 Endnotes	233
9	Feature Engineering and Selection	237
	9.1 Introduction	237
	9.2 Histogram Features	238
	9.3 Feature Scaling via Standard Normalization	249
	9.4 Imputing Missing Values in a Dataset	254
	9.5 Feature Scaling via PCA-Sphering	255
	9.6 Feature Selection via Boosting	258
	9.7 Feature Selection via Regularization	264
	9.8 Conclusion	269
	9.9 Exercises	269
Part III	Nonlinear Learning	273
10	Principles of Nonlinear Feature Engineering	275
	10.1 Introduction	275
	10.2 Nonlinear Regression	275
	10.3 Nonlinear Multi-Output Regression	282
	10.4 Nonlinear Two-Class Classification	286
	10.5 Nonlinear Multi-Class Classification	290
	10.6 Nonlinear Unsupervised Learning	294
	10.7 Conclusion	298
	10.8 Exercises	298
11	Principles of Feature Learning	304
	11.1 Introduction	304
	11.2 Universal Approximators	307
	11.3 Universal Approximation of Real Data	323
	11.4 Naive Cross-Validation	335
	11.5 Efficient Cross-Validation via Boosting	340
	11.6 Efficient Cross-Validation via Regularization	350
	11.7 Testing Data	361
	11.8 Which Universal Approximator Works Best in Practice?	365
	11.9 Bagging Cross-Validated Models	366

x	Contents	
	11.10 K-Fold Cross-Validation	373
	11.11 When Feature Learning Fails	378
	11.12 Conclusion	379
	11.13 Exercises	380
12	Kernel Methods	383
	12.1 Introduction	383
	12.2 Fixed-Shape Universal Approximators	383
	12.3 The Kernel Trick	386
	12.4 Kernels as Measures of Similarity	396
	12.5 Optimization of Kernelized Models	397
	12.6 Cross-Validating Kernelized Learners	398
	12.7 Conclusion	399
	12.8 Exercises	399
13	Fully Connected Neural Networks	403
	13.1 Introduction	403
	13.2 Fully Connected Neural Networks	403
	13.3 Activation Functions	424
	13.4 The Backpropagation Algorithm	427
	13.5 Optimization of Neural Network Models	428
	13.6 Batch Normalization	430
	13.7 Cross-Validation via Early Stopping	438
	13.8 Conclusion	440
	13.9 Exercises	441
14	Tree-Based Learners	443
	14.1 Introduction	443
	14.2 From Stumps to Deep Trees	443
	14.3 Regression Trees	446
	14.4 Classification Trees	452
	14.5 Gradient Boosting	458
	14.6 Random Forests	462
	14.7 Cross-Validation Techniques for Recursively Defined Trees	464
	14.8 Conclusion	467
	14.9 Exercises	467
Part IV	Appendices	471
Appendix A	Advanced First- and Second-Order Optimization Methods	473
	A.1 Introduction	473
	A.2 Momentum-Accelerated Gradient Descent	473
	A.3 Normalized Gradient Descent	478
	A.4 Advanced Gradient-Based Methods	485

A.5	Mini-Batch Optimization	487
A.6	Conservative Steplength Rules	490
A.7	Newton’s Method, Regularization, and Nonconvex Functions	499
A.8	Hessian-Free Methods	502
Appendix B	Derivatives and Automatic Differentiation	511
B.1	Introduction	511
B.2	The Derivative	511
B.3	Derivative Rules for Elementary Functions and Operations	514
B.4	The Gradient	516
B.5	The Computation Graph	517
B.6	The Forward Mode of Automatic Differentiation	520
B.7	The Reverse Mode of Automatic Differentiation	526
B.8	Higher-Order Derivatives	529
B.9	Taylor Series	531
B.10	Using the autograd Library	536
Appendix C	Linear Algebra	546
C.1	Introduction	546
C.2	Vectors and Vector Operations	546
C.3	Matrices and Matrix Operations	553
C.4	Eigenvalues and Eigenvectors	556
C.5	Vector and Matrix Norms	559
	<i>References</i>	564
	<i>Index</i>	569

Preface

For eons we humans have sought out *rules* or *patterns* that accurately describe how important systems in the world around us work, whether these systems be agricultural, biological, physical, financial, etc. We do this because such rules allow us to understand a system better, accurately predict its future behavior and ultimately, control it. However, the process of finding the “right” rule that seems to govern a given system has historically been no easy task. For most of our history *data* (glimpses of a given system at work) has been an extremely scarce commodity. Moreover, our ability to *compute*, to try out various rules to see which most accurately represents a phenomenon, has been limited to what we could accomplish by hand. Both of these factors naturally limited the range of phenomena scientific pioneers of the past could investigate and inevitably forced them to use philosophical and/or visual approaches to rule-finding. Today, however, we live in a world awash in data, and have colossal computing power at our fingertips. Because of this, we lucky descendants of the great pioneers can tackle a much wider array of problems and take a much more empirical approach to rule-finding than our forbears could. Machine learning, the topic of this textbook, is a term used to describe a broad (and growing) collection of pattern-finding algorithms designed to properly identify system rules empirically and by leveraging our access to potentially enormous amounts of data and computing power.

In the past decade the user base of machine learning has grown dramatically. From a relatively small circle in computer science, engineering, and mathematics departments the users of machine learning now include students and researchers from every corner of the academic universe, as well as members of industry, data scientists, entrepreneurs, and machine learning enthusiasts. This textbook is the result of a complete tearing down of the standard curriculum of machine learning into its most fundamental components, and a curated re-assembly of those pieces (painstakingly polished and organized) that we feel will most benefit this broadening audience of learners. It contains fresh and intuitive yet rigorous descriptions of the most fundamental concepts necessary to conduct research, build products, and tinker.

Book Overview

The second edition of this text is a complete revision of our first endeavor, with virtually every chapter of the original rewritten from the ground up and eight new chapters of material added, doubling the size of the first edition. Topics from the first edition, from expositions on gradient descent to those on One-versus-All classification and Principal Component Analysis have been reworked and polished. A swath of new topics have been added throughout the text, from derivative-free optimization to weighted supervised learning, feature selection, nonlinear feature engineering, boosting-based cross-validation, and more.

While heftier in size, the intent of our original attempt has remained unchanged: to explain machine learning, from first principles to practical implementation, in the simplest possible terms. A big-picture breakdown of the second edition text follows below.

Part I: Mathematical Optimization (Chapters 2–4)

Mathematical optimization is the workhorse of machine learning, powering not only the tuning of individual machine learning models (introduced in Part II) but also the framework by which we determine appropriate models themselves via cross-validation (discussed in Part III of the text).

In this first part of the text we provide a complete introduction to mathematical optimization, from basic zero-order (derivative-free) methods detailed in Chapter 2 to fundamental and advanced first-order and second-order methods in Chapters 3 and 4, respectively. More specifically this part of the text contains complete descriptions of local optimization, *random search* methodologies, *gradient descent*, and *Newton's method*.

Part II: Linear Learning (Chapters 5–9)

In this part of the text we describe the fundamental components of cost function based machine learning, with an emphasis on linear models.

This includes a complete description of *supervised learning* in Chapters 5–7 including linear regression, two-class, and multi-class classification. In each of these chapters we describe a range of perspectives and popular design choices made when building supervised learners.

In Chapter 8 we similarly describe *unsupervised learning*, and Chapter 9 contains an introduction to fundamental *feature engineering* practices including popular *histogram* features as well as various input normalization schemes, and *feature selection* paradigms.

Part III: Nonlinear Learning (Chapters 10–14)

In the final part of the text we extend the fundamental paradigms introduced in Part II to the general nonlinear setting.

We do this carefully beginning with a basic introduction to nonlinear supervised and unsupervised learning in Chapter 10, where we introduce the motivation, common terminology, and notation of nonlinear learning used throughout the remainder of the text.

In Chapter 11 we discuss how to *automate* the selection of appropriate nonlinear models, beginning with an introduction to *universal approximation*. This naturally leads to detailed descriptions of *cross-validation*, as well as *boosting*, *regularization*, *ensembling*, and *K-folds* cross-validation.

With these fundamental ideas in-hand, in Chapters 12–14 we then dedicate an individual chapter to each of the three popular universal approximators used in machine learning: *fixed-shape kernels*, *neural networks*, and *trees*, where we discuss the strengths, weaknesses, technical eccentricities, and usages of each popular universal approximator.

To get the most out of this part of the book we strongly recommend that Chapter 11 and the fundamental ideas therein are studied and understood before moving on to Chapters 12–14.

Part IV: Appendices

This shorter set of appendix chapters provides a complete treatment on advanced optimization techniques, as well as a thorough introduction to a range of subjects that the readers will need to understand in order to make full use of the text.

Appendix A continues our discussion from Chapters 3 and 4, and describes *advanced first- and second-order optimization techniques*. This includes a discussion of popular extensions of gradient descent, including *mini-batch optimization*, *momentum acceleration*, *gradient normalization*, and the result of combining these enhancements in various ways (producing e.g., the RMSProp and Adam first order algorithms) – and Newton’s method – including *regularization* schemes and *Hessian-free* methods.

Appendix B contains a tour of *computational calculus* including an introduction to the derivative/gradient, higher-order derivatives, the Hessian matrix, numerical differentiation, forward and backward (backpropagation) automatic differentiation, and Taylor series approximations.

Appendix C provides a suitable background in *linear and matrix algebra*, including vector/matrix arithmetic, the notions of spanning sets and orthogonality, as well as eigenvalues and eigenvectors.

Readers: How To Use This Book

This textbook was written with first-time learners of the subject in mind, as well as for more knowledgeable readers who yearn for a more intuitive and serviceable treatment than what is currently available today. To make full use of the text one needs only a basic understanding of vector algebra (mathematical functions, vector arithmetic, etc.) and computer programming (for example, basic proficiency with a dynamically typed language like Python). We provide complete introductory treatments of other prerequisite topics including linear algebra, vector calculus, and automatic differentiation in the appendices of the text. Example “roadmaps,” shown in Figures 0.1–0.4, provide suggested paths for navigating the text based on a variety of learning outcomes and university courses (ranging from a course on the essentials of machine learning to special topics – as described further under “Instructors: How to use this Book” below).

We believe that *intuitive leaps precede intellectual ones*, and to this end defer the use of probabilistic and statistical views of machine learning in favor of a fresh and consistent geometric perspective throughout the text. We believe that this perspective not only permits a more intuitive understanding of individual concepts in the text, but also that it helps establish revealing connections between ideas often regarded as fundamentally distinct (e.g., the logistic regression and Support Vector Machine classifiers, kernels and fully connected neural networks, etc.). We also highly emphasize the importance of *mathematical optimization* in our treatment of machine learning. As detailed in the “Book Overview” section above, optimization is the workhorse of machine learning and is fundamental at many levels – from the tuning of individual models to the general selection of appropriate nonlinearities via cross-validation. Because of this a strong understanding of mathematical optimization is requisite if one wishes to deeply understand machine learning, and if one wishes to be able to implement fundamental algorithms.

To this end, we place significant emphasis on the design and implementation of algorithms throughout the text with implementations of fundamental algorithms given in Python. These fundamental examples can then be used as building blocks for the reader to help complete the text’s programming exercises, allowing them to “get their hands dirty” and “learn by doing,” practicing the concepts introduced in the body of the text. While in principle any programming language can be used to complete the text’s coding exercises, we highly recommend using Python for its ease of use and large support community. We also recommend using the open-source Python libraries NumPy, autograd, and matplotlib, as well as the Jupyter notebook editor to make implementing and testing code easier. A complete set of installation instructions, datasets, as well as starter notebooks for many exercises can be found at

https://github.com/jermwatt/machine_learning_refined

Instructors: How To Use This Book

Chapter slides associated with this textbook, datasets, along with a large array of instructional interactive Python widgets illustrating various concepts throughout the text, can be found on the github repository accompanying this textbook at

https://github.com/jermwatt/machine_learning_refined

This site also contains instructions for installing Python as well as a number of other free packages that students will find useful in completing the text's exercises.

This book has been used as a basis for a number of machine learning courses at Northwestern University, ranging from introductory courses suitable for undergraduate students to more advanced courses on special topics focusing on optimization and deep learning for graduate students. With its treatment of foundations, applications, and algorithms this text can be used as a primary resource or in fundamental component for courses such as the following.

Machine learning essentials treatment: an introduction to the essentials of machine learning is ideal for undergraduate students, especially those in quarter-based programs and universities where a deep dive into the entirety of the book is not feasible due to time constraints. Topics for such a course can include: gradient descent, logistic regression, Support Vector Machines, One-versus-All and multi-class logistic regression, Principal Component Analysis, K-means clustering, the essentials of feature engineering and selection, cross-validation, regularization, ensembling, bagging, kernel methods, fully connected neural networks, and trees. A recommended roadmap for such a course – including recommended chapters, sections, and corresponding topics – is shown in Figure 0.1.

Machine learning full treatment: a standard machine learning course based on this text expands on the essentials course outlined above both in terms of breadth and depth. In addition to the topics mentioned in the essentials course, instructors may choose to cover Newton's method, Least Absolute Deviations, multi-output regression, weighted regression, the Perceptron, the Categorical Cross Entropy cost, weighted two-class and multi-class classification, online learning, recommender systems, matrix factorization techniques, boosting-based feature selection, universal approximation, gradient boosting, random forests, as well as a more in-depth treatment of fully connected neural networks involving topics such as batch normalization and early-stopping-based regularization. A recommended roadmap for such a course – including recommended chapters, sections, and corresponding topics – is illustrated in Figure 0.2.

Mathematical optimization for machine learning and deep learning: such a course entails a comprehensive description of zero-, first-, and second-order optimization techniques from Part I of the text (as well as Appendix A) including: coordinate descent, gradient descent, Newton’s method, quasi-Newton methods, stochastic optimization, momentum acceleration, fixed and adaptive steplength rules, as well as advanced normalized gradient descent schemes (e.g., Adam and RMSProp). These can be followed by an in-depth description of the feature engineering processes (especially standard normalization and PCA-sphering) that speed up (particularly first-order) optimization algorithms. All students in general, and those taking an optimization for machine learning course in particular, should appreciate the fundamental role optimization plays in identifying the “right” nonlinearity via the processes of boosting and regularization based cross-validation, the principles of which are covered in Chapter 11. Select topics from Chapter 13 and Appendix B – including backpropagation, batch normalization, and forward/backward mode of automatic differentiation – can also be covered. A recommended roadmap for such a course – including recommended chapters, sections, and corresponding topics – is given in Figure 0.3.

Introductory portion of a course on deep learning: such a course is best suitable for students who have had prior exposure to fundamental machine learning concepts, and can begin with a discussion of appropriate first order optimization techniques, with an emphasis on stochastic and mini-batch optimization, momentum acceleration, and normalized gradient schemes such as Adam and RMSProp. Depending on the audience, a brief review of fundamental elements of machine learning may be needed using selected portions of Part II of the text. A complete discussion of fully connected networks, including a discussion of backpropagation and forward/backward mode of automatic differentiation, as well as special topics like batch normalization and early-stopping-based cross-validation, can then be made using Chapters 11, 13, and Appendices A and B of the text. A recommended roadmap for such a course – including recommended chapters, sections, and corresponding topics – is shown in Figure 0.4. Additional recommended resources on topics to complete a standard course on deep learning – like convolutional and recurrent networks – can be found by visiting the text’s github repository.

CHAPTER	SECTIONS	TOPICS
1	1 2 3 4 5	Machine Learning Taxonomy
2	1 2 3 4 5	Global/Local Optimization Curse of Dimensionality
3	1 2 3 4 5	Gradient Descent
4		
5	1 2	Least Squares Linear Regression
6	1 2 3 5 6 8	Logistic Regression Cross Entropy/Softmax Cost SVMs
7	1 2 3 4 6	One-versus-All Multi-Class Logistic Regression
8	1 2 3 5	Principal Component Analysis K-means
9	2 7	Feature Engineering Feature Selection
10	1 2 4	Nonlinear Regression Nonlinear Classification
11	1 2 3 4 6 7 9	Universal Approximation Cross-Validation Regularization
		Ensembling Bagging
12	1 2 3	Kernel Methods The Kernel Trick
13	1 2 4	Fully Connected Networks Backpropagation
14	1 2 3 4	Regression Trees Classification Trees
A		
B		
C		

Figure 0.1 Recommended study roadmap for a course on the essentials of machine learning, including requisite chapters (left column), sections (middle column), and corresponding topics (right column). This essentials plan is suitable for time-constrained courses (in quarter-based programs and universities) or self-study, or where machine learning is not the sole focus but a key component of some broader course of study. Note that chapters are grouped together visually based on text layout detailed under “Book Overview” in the Preface. See the section titled “Instructors: How To Use This Book” in the Preface for further details.

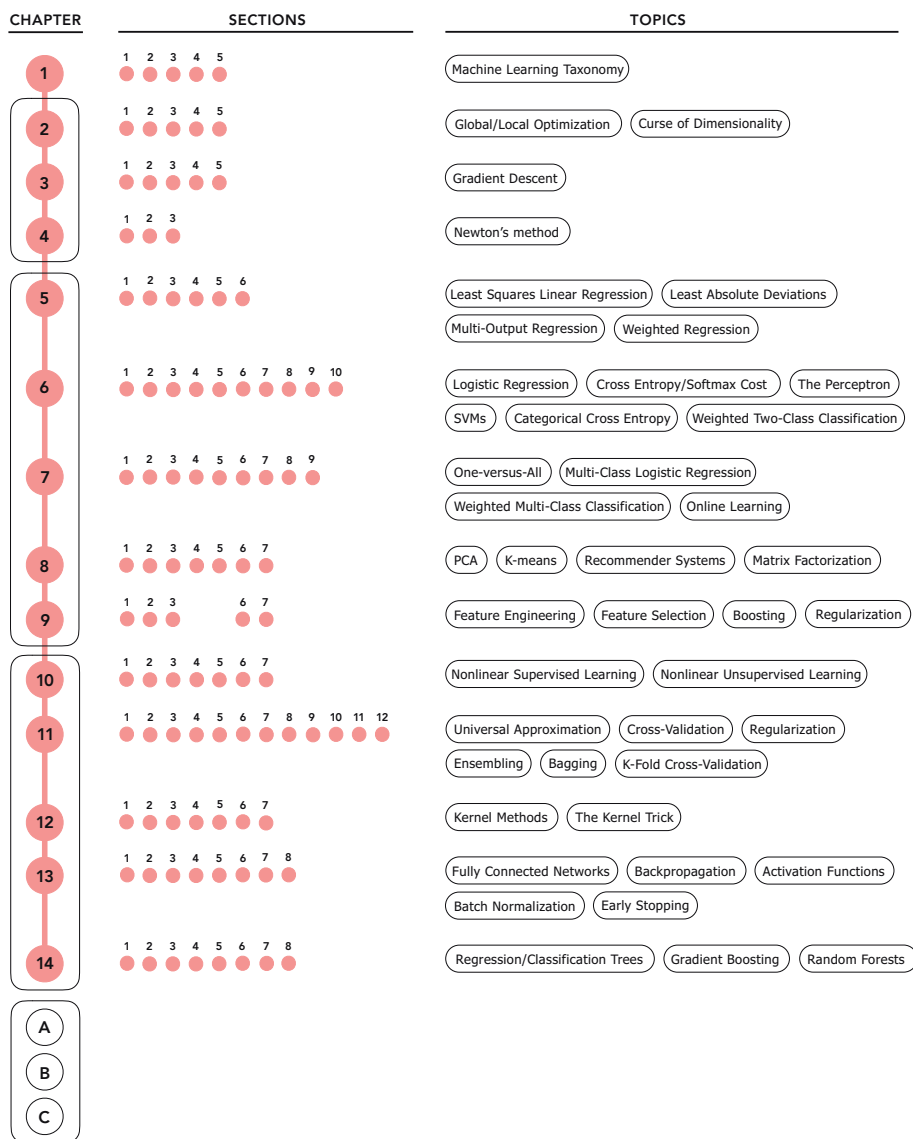


Figure 0.2 Recommended study roadmap for a full treatment of standard machine learning subjects, including chapters, sections, as well as corresponding topics to cover. This plan entails a more in-depth coverage of machine learning topics compared to the essentials roadmap given in Figure 0.1, and is best suited for senior undergraduate/early graduate students in semester-based programs and passionate independent readers. See the section titled “Instructors: How To Use This Book” in the Preface for further details.

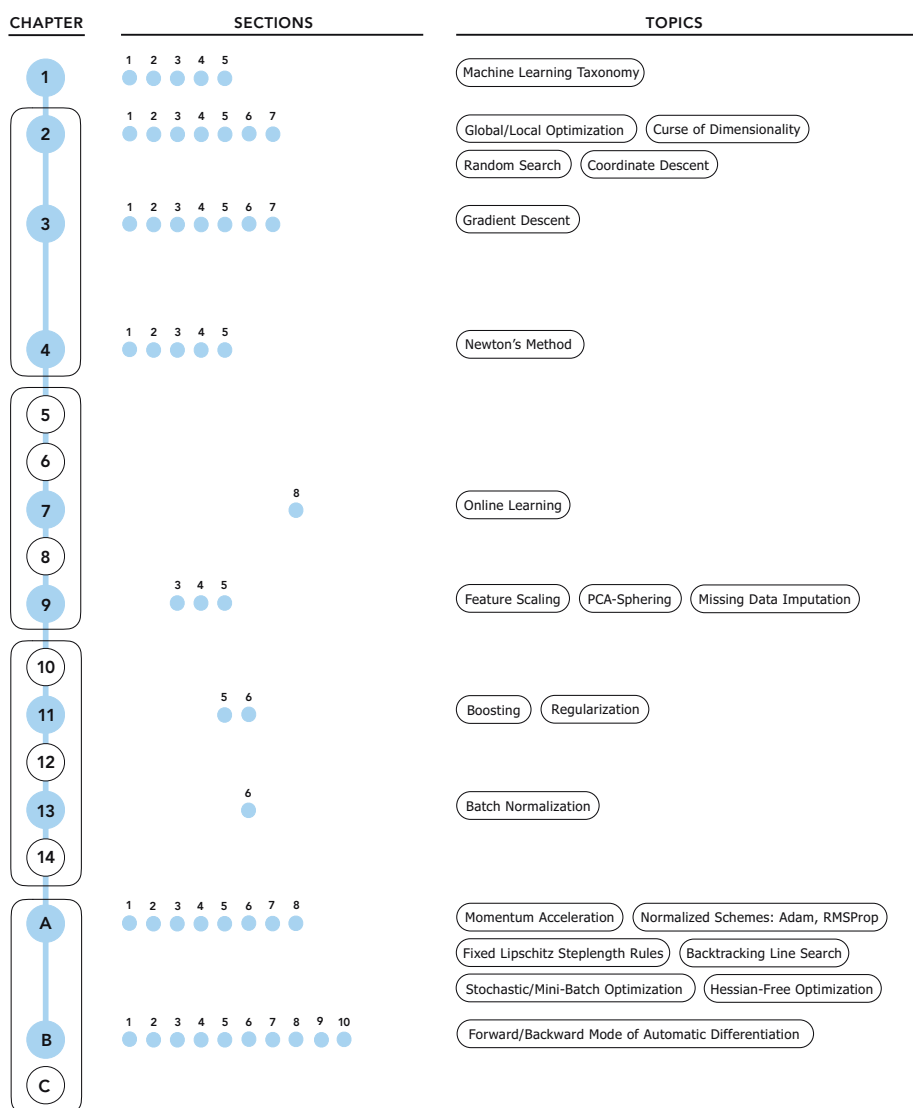


Figure 0.3 Recommended study roadmap for a course on mathematical optimization for machine learning and deep learning, including chapters, sections, as well as topics to cover. See the section titled “Instructors: How To Use This Book” in the Preface for further details.

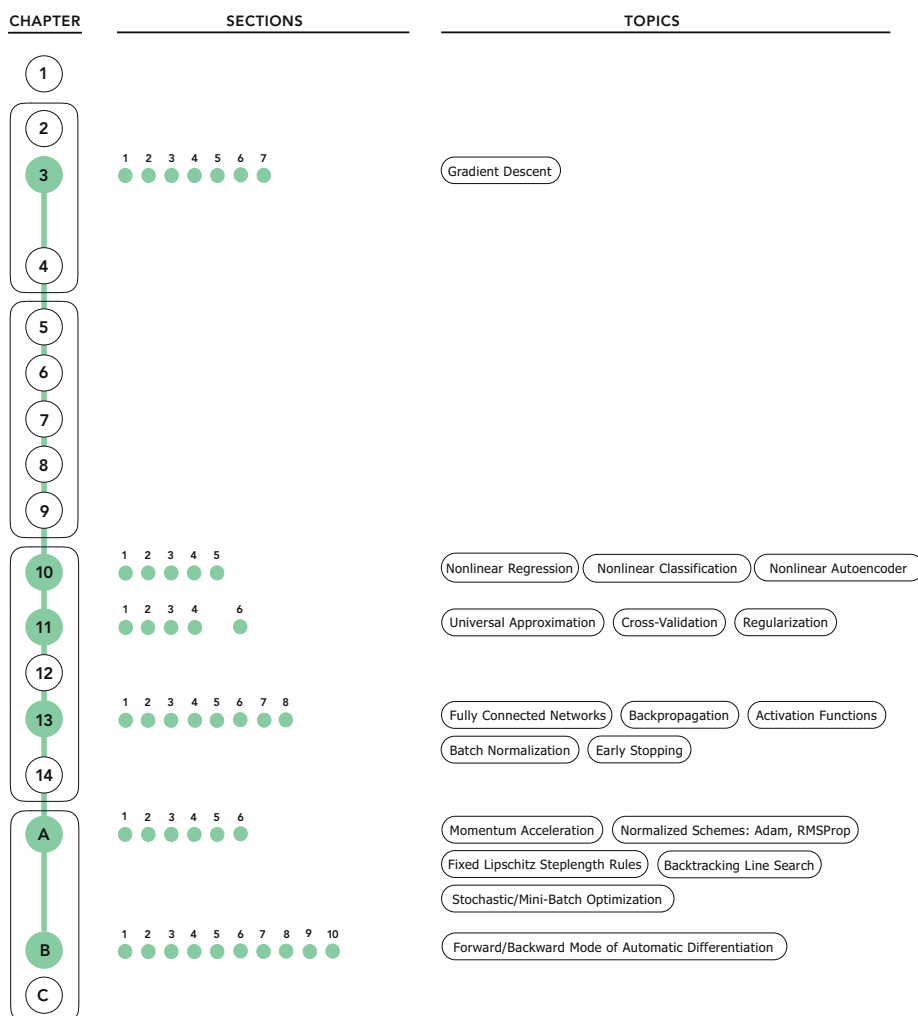


Figure 0.4 Recommended study roadmap for an introductory portion of a course on deep learning, including chapters, sections, as well as topics to cover. See the section titled “Instructors: How To Use This Book” in the Preface for further details.

Acknowledgements

This text could not have been written in anything close to its current form without the enormous work of countless genius-angels in the Python open-source community, particularly authors and contributors of NumPy, Jupyter, and matplotlib. We are especially grateful to the authors and contributors of autograd including Dougal Maclaurin, David Duvenaud, Matt Johnson, and Jamie Townsend, as autograd allowed us to experiment and iterate on a host of new ideas included in the second edition of this text that greatly improved it as well as, we hope, the learning experience for its readers.

We are also very grateful for the many students over the years that provided insightful feedback on the content of this text, with special thanks to Bowen Tian who provided copious amounts of insightful feedback on early drafts of the work.

Finally, a big thanks to Mark McNess Rosengren and the entire Standing Passengers crew for helping us stay caffeinated during the writing of this text.