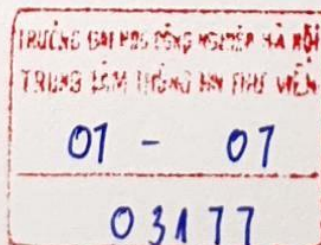


INTRODUCTION TO **DEEP LEARNING**

EUGENE CHARNIAK

Introduction to Deep Learning

Eugene Charniak



The MIT Press
Cambridge, Massachusetts
London, England

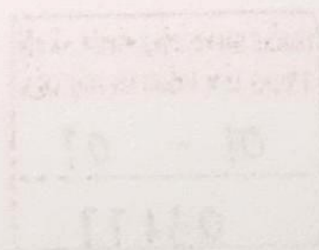
© 2018 The Massachusetts Institute of Technology

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

This book was set in L^AT_EX by author. Printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data is available.
ISBN: 978-0-262-03951-2

10 9 8 7 6 5 4 3 2 1



Contents

Preface	xi
1 Feed-Forward Neural Nets	1
1.1 Perceptrons	3
1.2 Cross-entropy Loss Functions for Neural Nets	9
1.3 Derivatives and Stochastic Gradient Descent	14
1.4 Writing Our Program	18
1.5 Matrix Representation of Neural Nets	21
1.6 Data Independence	24
1.7 References and Further Readings	25
1.8 Written Exercises	26
2 Tensorflow	29
2.1 Tensorflow Preliminaries	29
2.2 A TF Program	33
2.3 Multilayered NNs	38
2.4 Other Pieces	42
2.4.1 Checkpointing	42
2.4.2 <code>tf.nn.matmul</code>	43
2.4.3 Initialization of TF Variables	44
2.4.4 Simplifying TF Graph Creation	47
2.5 References and Further Readings	48
2.6 Written Exercises	49
3 Convolutional Neural Networks	51
3.1 Filters, Strides, and Padding	52
3.2 A Simple TF Convolution Example	57
3.3 Multilevel Convolution	61
3.4 Convolution Details	64

3.4.1	Biases	64
3.4.2	Layers with Convolution	65
3.4.3	Pooling	66
3.5	References and Further Readings	67
3.6	Written Exercises	68
4	Word Embeddings and Recurrent NNs	71
4.1	Word Embeddings for Language Models	71
4.2	Building Feed-Forward Language Models	76
4.3	Improving Feed-Forward Language Models	78
4.4	Overfitting	79
4.5	Recurrent Networks	82
4.6	Long Short-Term Memory	88
4.7	References and Further Readings	92
4.8	Written Exercises	92
5	Sequence-to-Sequence Learning	95
5.1	The Seq2Seq Paradigm	96
5.2	Writing a Seq2Seq MT program	99
5.3	Attention in Seq2seq	102
5.4	Multilength Seq2Seq	107
5.5	Programming Exercise	108
5.6	Written Exercises	110
5.7	References and Further Readings	111
6	Deep Reinforcement Learning	113
6.1	Value Iteration	114
6.2	Q-learning	117
6.3	Basic Deep-Q Learning	119
6.4	Policy Gradient Methods	124
6.5	Actor-Critic Methods	130
6.6	Experience Replay	133
6.7	References and Further Readings	134
6.8	Written Exercises	134
7	Unsupervised Neural-Network Models	137
7.1	Basic Autoencoding	137
7.2	Convolutional Autoencoding	140
7.3	Variational Autoencoding	144
7.4	Generative Adversarial Networks	152

7.5	References and Further Readings	157
7.6	Written Exercises	157
A	Answers to Selected Exercises	159
A.1	Chapter 1	159
A.2	Chapter 2	160
A.3	Chapter 3	160
A.4	Chapter 4	161
A.5	Chapter 5	161
A.6	Chapter 6	162
A.7	Chapter 7	162
	Bibliography	165
	Index	169

Preface

Your author is a long-time artificial-intelligence researcher whose field of expertise, natural-language processing, has been revolutionized by deep learning. Unfortunately, it took him (me) a long time to catch on to this fact. I can rationalize this since this is the third time neural networks have threatened a revolution but only the first time they have delivered. Nevertheless, I suddenly found myself way behind the times and struggling to catch up. So I did what any self-respecting professor would do, scheduled myself to teach the stuff, started a crash course by surfing the web, and got my students to teach it to me. (This last is not a joke. In particular, the head undergraduate teaching assistant for the course, Siddarth (Sidd) Karramcheti, deserves special mention.)

This explains several prominent features of this book. First, it is short. I am a slow learner. Second, it is very much project driven. Many texts, particularly in computer science, have a constant tension between topic organization and organizing material around specific projects. Splitting the difference is often a good idea, but I find I learn computer science material best by sitting down and writing programs, so my book largely reflects my learning habits. It was the most convenient way to put it down, and I am hoping many in the expected audience will find it helpful as well.

Which brings up the question of the expected audience. While I hope many CS practitioners will find this book useful for the same reason I wrote it, as a teacher my first loyalty is to my students, so this book is primarily intended as a textbook for a course on deep learning. The course I teach at Brown is for both graduate and undergraduates and covers all the material herein, plus some “culture” lectures (for graduate credit a student must add a significant final project). Both linear algebra and multivariate calculus are required. While the actual quantity of linear-algebra material is not that great, students have told me that without it they would have found thinking about multilevel networks, and the tensors they require, quite difficult. Multivariate calculus, however, was a much closer call. It appears explicitly

only in Chapter 1, when we build up to back-propagation from scratch and I would not be surprised if an extra lecture on partial derivatives would do. Last, there is a probability and statistics prerequisite. This simplifies the exposition and I certainly want to encourage students to take such a course. I also assume a rudimentary knowledge of programming in Python. I do not include this in the text, but my course has an extra “lab” on basic Python.

That your author was playing catch-up when writing this book also explains the fact that in almost every chapter’s section on further reading you will find, beyond the usual references to important research papers, many reference to secondary sources — others’ educational writings. I would never have learned this material without them.

Providence, Rhode Island
January 2018