



contents

preface	xv
acknowledgments	xvi
about this book	xvii
about the author	xx
1 Introducing deep learning: why you should learn it	3
Welcome to <i>Grokking Deep Learning</i>	3
Why you should learn deep learning	4
Will this be difficult to learn?	5
Why you should read this book	5
What you need to get started	7
You'll probably need some Python knowledge	8
Summary	8
2 Fundamental concepts: how do machines learn?	9
What is deep learning?	10
What is machine learning?	11
Supervised machine learning	12
Unsupervised machine learning	13
Parametric vs. nonparametric learning	14
Supervised parametric learning	15
Unsupervised parametric learning	17
Nonparametric learning	18
Summary	19

3 Introduction to neural prediction: forward propagation	21
Step 1: Predict	22
A simple neural network making a prediction	24
What is a neural network?	25
What does this neural network do?	26
Making a prediction with multiple inputs	28
Multiple inputs: What does this neural network do?	30
Multiple inputs: Complete runnable code	35
Making a prediction with multiple outputs	36
Predicting with multiple inputs and outputs	38
Multiple inputs and outputs: How does it work?	40
Predicting on predictions	42
A quick primer on NumPy	44
Summary	46
4 Introduction to neural learning: gradient descent	47
Predict, compare, and learn	48
Compare	48
Learn	49
Compare: Does your network make good predictions?	50
Why measure error?	51
What's the simplest form of neural learning?	52
Hot and cold learning	54
Characteristics of hot and cold learning	55
Calculating both direction and amount from error	56
One iteration of gradient descent	58
Learning is just reducing error	60
Let's watch several steps of learning	62
Why does this work? What is weight_delta, really?	64
Tunnel vision on one concept	66
A box with rods poking out of it	67
Derivatives: Take two	68
What you really need to know	69
What you don't really need to know	69
How to use a derivative to learn	70
Look familiar?	71

Breaking gradient descent	72
Visualizing the overcorrections	73
Divergence	74
Introducing alpha	75
Alpha in code	76
Memorizing	77

5 Learning multiple weights at a time: generalizing gradient descent 79

Gradient descent learning with multiple inputs	80
Gradient descent with multiple inputs explained	82
Let's watch several steps of learning	86
Freezing one weight: What does it do?	88
Gradient descent learning with multiple outputs	90
Gradient descent with multiple inputs and outputs	92
What do these weights learn?	94
Visualizing weight values	96
Visualizing dot products (weighted sums)	97
Summary	98

6 Building your first deep neural network: introduction to backpropagation 99

The streetlight problem	100
Preparing the data	102
Matrices and the matrix relationship	103
Creating a matrix or two in Python	106
Building a neural network	107
Learning the whole dataset	108
Full, batch, and stochastic gradient descent	109
Neural networks learn correlation	110
Up and down pressure	111
Edge case: Overfitting	113
Edge case: Conflicting pressure	114
Learning indirect correlation	116
Creating correlation	117
Stacking neural networks: A review	118
Backpropagation: Long-distance error attribution	119

Backpropagation: Why does this work?	120
Linear vs. nonlinear	121
Why the neural network still doesn't work	122
The secret to sometimes correlation	123
A quick break	124
Your first deep neural network	125
Backpropagation in code	126
One iteration of backpropagation	128
Putting it all together	130
Why do deep networks matter?	131
7 How to picture neural networks: in your head and on paper	133
.....	
It's time to simplify	134
Correlation summarization	135
The previously overcomplicated visualization	136
The simplified visualization	137
Simplifying even further	138
Let's see this network predict	139
Visualizing using letters instead of pictures	140
Linking the variables	141
Everything side by side	142
The importance of visualization tools	143
8 Learning signal and ignoring noise: introduction to regularization and batching	145
.....	
Three-layer network on MNIST	146
Well, that was easy	148
Memorization vs. generalization	149
Overfitting in neural networks	150
Where overfitting comes from	151
The simplest regularization: Early stopping	152
Industry standard regularization: Dropout	153
Why dropout works: Ensembling works	154
Dropout in code	155
Dropout evaluated on MNIST	157
Batch gradient descent	158
Summary	160

9 Modeling probabilities and nonlinearities: activation functions	161
.....	
What is an activation function?	162
Standard hidden-layer activation functions	165
Standard output layer activation functions	166
The core issue: Inputs have similarity	168
softmax computation	169
Activation installation instructions	170
Multiplying delta by the slope	172
Converting output to slope (derivative)	173
Upgrading the MNIST network	174
10 Neural learning about edges and corners: intro to convolutional neural networks	177
.....	
Reusing weights in multiple places	178
The convolutional layer	179
A simple implementation in NumPy	181
Summary	185
11 Neural networks that understand language: king – man + woman == ?	187
.....	
What does it mean to understand language?	188
Natural language processing (NLP)	189
Supervised NLP	190
IMDB movie reviews dataset	191
Capturing word correlation in input data	192
Predicting movie reviews	193
Intro to an embedding layer	194
Interpreting the output	196
Neural architecture	197
Comparing word embeddings	199
What is the meaning of a neuron?	200
Filling in the blank	201
Meaning is derived from loss	203
King – Man + Woman \approx Queen	206
Word analogies	207
Summary	208

12 Neural networks that write like Shakespeare: recurrent layers for variable-length data	209
.....	
The challenge of arbitrary length	210
Do comparisons really matter?	211
The surprising power of averaged word vectors	212
How is information stored in these embeddings?	213
How does a neural network use embeddings?	214
The limitations of bag-of-words vectors	215
Using identity vectors to sum word embeddings	216
Matrices that change absolutely nothing	217
Learning the transition matrices	218
Learning to create useful sentence vectors	219
Forward propagation in Python	220
How do you backpropagate into this?	221
Let's train it!	222
Setting things up	223
Forward propagation with arbitrary length	224
Backpropagation with arbitrary length	225
Weight update with arbitrary length	226
Execution and output analysis	227
Summary	229
13 Introducing automatic optimization: let's build a deep learning framework	231
.....	
What is a deep learning framework?	232
Introduction to tensors	233
Introduction to automatic gradient computation (autograd)	234
A quick checkpoint	236
Tensors that are used multiple times	237
Upgrading autograd to support multiuse tensors	238
How does addition backpropagation work?	240
Adding support for negation	241
Adding support for additional functions	242
Using autograd to train a neural network	246
Adding automatic optimization	248
Adding support for layer types	249

Layers that contain layers	250
Loss-function layers	251
How to learn a framework	252
Nonlinearity layers	253
The embedding layer	255
Adding indexing to autograd	256
The embedding layer (revisited)	257
The cross-entropy layer	258
The recurrent neural network layer	260
Summary	263
14 Learning to write like Shakespeare: long short-term memory	265
.....
Character language modeling	266
The need for truncated backpropagation	267
Truncated backpropagation	268
A sample of the output	271
Vanishing and exploding gradients	272
A toy example of RNN backpropagation	273
Long short-term memory (LSTM) cells	274
Some intuition about LSTM gates	275
The long short-term memory layer	276
Upgrading the character language model	277
Training the LSTM character language model	278
Tuning the LSTM character language model	279
Summary	280
15 Deep learning on unseen data: introducing federated learning	281
.....
The problem of privacy in deep learning	282
Federated learning	283
Learning to detect spam	284
Let's make it federated	286
Hacking into federated learning	287
Secure aggregation	288
Homomorphic encryption	289

Homomorphically encrypted federated learning	290
Summary	291
16 Where to go from here: a brief guide	293
.....	
Congratulations!	294
Step 1: Start learning PyTorch	294
Step 2: Start another deep learning course	295
Step 3: Grab a mathy deep learning textbook	295
Step 4: Start a blog, and teach deep learning	296
Step 5: Twitter	297
Step 6: Implement academic papers	297
Step 7: Acquire access to a GPU (or many)	297
Step 8: Get paid to practice	298
Step 9: Join an open source project	298
Step 10: Develop your local community	299
index	301