
Contents

1	An Introduction to Neural Networks	1
1.1	Introduction	1
1.2	Single Computational Layer: The Perceptron	5
1.2.1	Use of Bias	8
1.2.2	What Objective Function Is the Perceptron Optimizing?	8
1.3	The Base Components of Neural Architectures	10
1.3.1	Choice of Activation Function	10
1.3.2	Softmax Activation Function	12
1.3.3	Common Loss Functions	13
1.4	Multilayer Neural Networks	13
1.4.1	The Multilayer Network as a Computational Graph	15
1.5	The Importance of Nonlinearity	17
1.5.1	Nonlinear Activations in Action	18
1.6	Advanced Architectures and Structured Data	20
1.7	Two Notable Benchmarks	21
1.7.1	The MNIST Database of Handwritten Digits	21
1.7.2	The ImageNet Database	22
1.8	Summary	23
1.9	Bibliographic Notes and Software Resources	23
1.10	Exercises	25
2	The Backpropagation Algorithm	29
2.1	Introduction	29
2.2	The Computational Graph Abstraction	30
2.2.1	Computational Graphs Create Complex Functions	31
2.3	Backpropagation in Computational Graphs	33
2.3.1	Computing Node-to-Node Derivatives with the Chain Rule . . .	34
2.3.2	Dynamic Programming for Computing Node-to-Node Derivatives	38
2.3.3	Converting Node-to-Node Derivatives into Loss-to-Weight Derivatives	42

2.4	Backpropagation in Neural Networks	44
2.4.1	Some Useful Derivatives of Activation Functions	46
2.4.2	Examples of Updates for Various Activations	48
2.5	The Vector-Centric View of Backpropagation	50
2.5.1	Derivatives with Respect to Vectors	51
2.5.2	Vector-Centric Chain Rule	51
2.5.3	A Decoupled View of Vector-Centric Backpropagation	52
2.5.4	Vector-Centric Backpropagation with Non-Layered Architectures	57
2.6	The Not-So-Unimportant Details	58
2.6.1	Mini-Batch Stochastic Gradient Descent	58
2.6.2	Learning Rate Decay	60
2.6.3	Checking the Correctness of Gradient Computation	60
2.6.4	Regularization	61
2.6.5	Loss Functions on Hidden Nodes	61
2.6.6	Backpropagation Tricks for Handling Shared Weights	62
2.7	Tuning and Preprocessing	62
2.7.1	Tuning Hyperparameters	63
2.7.2	Feature Preprocessing	64
2.7.3	Initialization	66
2.8	Backpropagation Is Interpretable	67
2.9	Summary	67
2.10	Bibliographic Notes and Software Resources	68
2.11	Exercises	68

Machine Learning with Shallow Neural Networks 73

3.1	Introduction	73
3.2	Neural Architectures for Binary Classification Models	75
3.2.1	Revisiting the Perceptron	75
3.2.2	Least-Squares Regression	76
3.2.2.1	Widrow-Hoff Learning	78
3.2.2.2	Closed Form Solutions	79
3.2.3	Support Vector Machines	79
3.2.4	Logistic Regression	81
3.2.5	Comparison of Different Models	82
3.3	Neural Architectures for Multiclass Models	84
3.3.1	Multiclass Perceptron	84
3.3.2	Weston-Watkins SVM	85
3.3.3	Multinomial Logistic Regression (Softmax Classifier)	86
3.4	Unsupervised Learning with Autoencoders	88
3.4.1	Linear Autoencoder with a Single Hidden Layer	89
3.4.1.1	Connections with Singular Value Decomposition	91
3.4.1.2	Sharing Weights in the Encoder and Decoder	91
3.4.2	Nonlinear Activation Functions and Depth	92
3.4.3	Application to Visualization	93
3.4.4	Application to Outlier Detection	95
3.4.5	Application to Multimodal Embeddings	95
3.4.6	Benefits of Autoencoders	96
3.5	Recommender Systems	96

3.6	Text Embedding with Word2vec	99
3.6.1	Neural Embedding with Continuous Bag of Words	100
3.6.2	Neural Embedding with Skip-Gram Model	103
3.6.3	Word2vec (SGNS) is Logistic Matrix Factorization	107
3.7	Simple Neural Architectures for Graph Embeddings	110
3.7.1	Handling Arbitrary Edge Counts	111
3.7.2	Beyond One-Hop Structural Models	112
3.7.3	Multinomial Model	112
3.8	Summary	113
3.9	Bibliographic Notes and Software Resources	113
3.10	Exercises	114

4 Deep Learning: Principles and Training Algorithms 119

4.1	Introduction	119
4.2	Why Is Depth Beneficial?	120
4.2.1	Hierarchical Feature Engineering: How Depth Reveals Rich Structure	120
4.3	Why Is Training Deep Networks Hard?	122
4.3.1	Geometric Understanding of the Effect of Gradient Ratios	122
4.3.2	The Vanishing and Exploding Gradient Problems	124
4.3.3	Cliffs and Valleys	126
4.3.4	Convergence Problems with Depth	127
4.3.5	Local Minima	127
4.4	Depth-Friendly Neural Architectures	129
4.4.1	Activation Function Choice	129
4.4.2	Dying Neurons and “Brain Damage”	130
4.4.2.1	Leaky ReLU	130
4.4.2.2	Maxout Networks	131
4.4.3	Using Skip Connections	131
4.5	Depth-Friendly Gradient-Descent Strategies	132
4.5.1	Importance of Preprocessing and Initialization	132
4.5.2	Momentum-Based Learning	133
4.5.3	Nesterov Momentum	134
4.5.4	Parameter-Specific Learning Rates	135
4.5.4.1	AdaGrad	136
4.5.4.2	RMSProp	136
4.5.4.3	AdaDelta	137
4.5.5	Combining Parameter-Specific Learning and Momentum	138
4.5.5.1	RMSProp with Nesterov Momentum	138
4.5.5.2	Adam	138
4.5.6	Gradient Clipping	139
4.5.7	Polyak Averaging	139
4.6	Second-Order Derivatives: The Newton Method	140
4.6.1	Example: Newton Method in the Quadratic Bowl	142
4.6.2	Example: Newton Method in a Non-Quadratic Function	142
4.6.3	The Saddle-Point Problem with Second-Order Methods	143
4.7	Fast Approximations of Newton Method	145
4.7.1	Conjugate Gradient Method	145
4.7.2	Quasi-Newton Methods and BFGS	148

8	Batch Normalization	150
9	Practical Tricks for Acceleration and Compression	153
	4.9.1 GPU Acceleration	154
	4.9.2 Parallel and Distributed Implementations	156
	4.9.3 Algorithmic Tricks for Model Compression	157
10	Summary	160
11	Bibliographic Notes and Software Resources	160
12	Exercises	162
	Teaching Deep Learners to Generalize	165
1	Introduction	165
	5.1.1 Example: Linear Regression	166
	5.1.2 Example: Polynomial Regression	167
2	The Bias-Variance Trade-Off	171
3	Generalization Issues in Model Tuning and Evaluation	174
	5.3.1 Evaluating with Hold-Out and Cross-Validation	176
	5.3.2 Issues with Training at Scale	177
	5.3.3 How to Detect Need to Collect More Data	178
4	Penalty-Based Regularization	178
	5.4.1 Connections with Noise Injection	179
	5.4.2 L_1 -Regularization	180
	5.4.3 L_1 - or L_2 -Regularization?	181
	5.4.4 Penalizing Hidden Units: Learning Sparse Representations	181
5.5	Ensemble Methods	182
	5.5.1 Bagging and Subsampling	182
	5.5.2 Parametric Model Selection and Averaging	184
	5.5.3 Randomized Connection Dropping	184
	5.5.4 Dropout	185
	5.5.5 Data Perturbation Ensembles	187
5.6	Early Stopping	188
	5.6.1 Understanding Early Stopping from the Variance Perspective	189
5.7	Unsupervised Pretraining	189
	5.7.1 Variations of Unsupervised Pretraining	192
	5.7.2 What About Supervised Pretraining?	193
5.8	Continuation and Curriculum Learning	194
5.9	Parameter Sharing	196
5.10	Regularization in Unsupervised Applications	197
	5.10.1 When the Hidden Layer is Broader than the Input Layer	197
	5.10.1.1 Sparse Feature Learning	198
	5.10.2 Noise Injection: De-noising Autoencoders	198
	5.10.3 Gradient-Based Penalization: Contractive Autoencoders	199
	5.10.4 Hidden Probabilistic Structure: Variational Autoencoders	203
	5.10.4.1 Reconstruction and Generative Sampling	206
	5.10.4.2 Conditional Variational Autoencoders	208
	5.10.4.3 Relationship with Generative Adversarial Networks	208
5.11	Summary	209
5.12	Bibliographic Notes and Software Resources	210
5.13	Exercises	211

6	Radial Basis Function Networks	215
6.1	Introduction	215
6.2	Training an RBF Network	218
	6.2.1 Training the Hidden Layer	218
	6.2.2 Training the Output Layer	220
	6.2.3 Iterative Construction of Hidden Layer	221
	6.2.4 Fully Supervised Learning of Hidden Layer	222
6.3	Variations and Special Cases of RBF Networks	223
	6.3.1 Classification with Perceptron Criterion	224
	6.3.2 Classification with Hinge Loss	224
	6.3.3 Example of Linear Separability Promoted by RBF	224
	6.3.4 Application to Interpolation	226
6.4	Relationship with Kernel Methods	227
	6.4.1 Kernel Regression Is a Special Case of RBF Networks	227
	6.4.2 Kernel SVM Is a Special Case of RBF Networks	228
6.5	Summary	229
6.6	Bibliographic Notes and Software Resources	229
6.7	Exercises	229
7	Restricted Boltzmann Machines	231
7.1	Introduction	231
7.2	Hopfield Networks	232
	7.2.1 Training a Hopfield Network	235
	7.2.2 Building a Toy Recommender and Its Limitations	236
	7.2.3 Increasing the Expressive Power of the Hopfield Network	237
7.3	The Boltzmann Machine	238
	7.3.1 How a Boltzmann Machine Generates Data	240
	7.3.2 Learning the Weights of a Boltzmann Machine	240
7.4	Restricted Boltzmann Machines	242
	7.4.1 Training the RBM	244
	7.4.2 Contrastive Divergence Algorithm	245
7.5	Applications of Restricted Boltzmann Machines	247
	7.5.1 Dimensionality Reduction and Data Reconstruction	247
	7.5.2 RBMs for Collaborative Filtering	249
	7.5.3 Using RBMs for Classification	252
	7.5.4 Topic Models with RBMs	254
	7.5.5 RBMs for Machine Learning with Multimodal Data	256
7.6	Using RBMs beyond Binary Data Types	258
7.7	Stacking Restricted Boltzmann Machines	258
	7.7.1 Unsupervised Learning	261
	7.7.2 Supervised Learning	261
	7.7.3 Deep Boltzmann Machines and Deep Belief Networks	261
7.8	Summary	262
7.9	Bibliographic Notes and Software Resources	262
7.10	Exercises	264

urrent Neural Networks		265
Introduction		265
The Architecture of Recurrent Neural Networks		267
8.2.1	Language Modeling Example of RNN	270
8.2.2	Backpropagation Through Time	273
8.2.3	Bidirectional Recurrent Networks	275
8.2.4	Multilayer Recurrent Networks	277
The Challenges of Training Recurrent Networks		278
8.3.1	Layer Normalization	281
Echo-State Networks		282
Long Short-Term Memory (LSTM)		285
Gated Recurrent Units (GRUs)		287
Applications of Recurrent Neural Networks		289
8.7.1	Contextualized Word Embeddings with ELMo	290
8.7.2	Application to Automatic Image Captioning	291
8.7.3	Sequence-to-Sequence Learning and Machine Translation	292
8.7.4	Application to Sentence-Level Classification	295
8.7.5	Token-Level Classification with Linguistic Features	296
8.7.6	Time-Series Forecasting and Prediction	297
8.7.7	Temporal Recommender Systems	299
8.7.8	Secondary Protein Structure Prediction	301
8.7.9	End-to-End Speech Recognition	301
8.7.10	Handwriting Recognition	301
8	Summary	302
9	Bibliographic Notes and Software Resources	302
10	Exercises	303
onvolutional Neural Networks		305
Introduction		305
9.1.1	Historical Perspective and Biological Inspiration	305
9.1.2	Broader Observations about Convolutional Neural Networks	306
9.2	The Basic Structure of a Convolutional Network	307
9.2.1	Padding	312
9.2.2	Strides	313
9.2.3	The ReLU Layer	315
9.2.4	Pooling	315
9.2.5	Fully Connected Layers	317
9.2.6	The Interleaving between Layers	317
9.2.7	Hierarchical Feature Engineering	320
9.3	Training a Convolutional Network	321
9.3.1	Backpropagating Through Convolutions	321
9.3.2	Backpropagation as Convolution with Inverted/Transposed Filter	322
9.3.3	Convolution/Backpropagation as Matrix Multiplications	324
9.3.4	Data Augmentation	326
9.4	Case Studies of Convolutional Architectures	326
9.4.1	AlexNet	327
9.4.2	ZFNet	329
9.4.3	VGG	330

9.4.4	GoogLeNet	333
9.4.5	ResNet	335
9.4.6	Squeeze-and-Excitation Networks (SENet)	338
9.4.7	The Effects of Depth	339
9.4.8	Pretrained Models	340
9.5	Visualization and Unsupervised Learning	341
9.5.1	Visualizing the Features of a Trained Network	341
9.5.2	Convolutional Autoencoders	347
9.6	Applications of Convolutional Networks	351
9.6.1	Content-Based Image Retrieval	352
9.6.2	Object Localization	352
9.6.3	Object Detection	354
9.6.4	Natural Language and Sequence Learning with TextCNN	355
9.6.5	Video Classification	355
9.7	Summary	356
9.8	Bibliographic Notes and Software Resources	356
9.9	Exercises	359
10 Graph Neural Networks		361
10.1	Introduction	361
10.2	Node Embeddings with Conventional Architectures	362
10.2.1	Adjacency Matrix Representation and Feature Engineering	364
10.3	Graph Neural Networks: The General Framework	364
10.3.1	The Neighborhood Function	368
10.3.2	Graph Convolution Function	368
10.3.3	GraphSAGE	369
10.3.4	Handling Edge Weights	371
10.3.5	Handling New Vertices	371
10.3.6	Handling Relational Networks	372
10.3.7	Directed Graphs	373
10.3.8	Gated Graph Neural Networks	373
10.3.9	Comparison with Image Convolutional Networks	374
10.4	Backpropagation in Graph Neural Networks	375
10.5	Beyond Nodes: Generating Graph-Level Models	377
10.6	Applications of Graph Neural Networks	382
10.7	Summary	384
10.8	Bibliographic Notes and Software Resources	384
10.9	Exercises	385
11 Deep Reinforcement Learning		389
11.1	Introduction	389
11.2	Stateless Algorithms: Multi-Armed Bandits	391
11.3	The Basic Framework of Reinforcement Learning	393
11.4	Monte Carlo Sampling	395
11.4.1	Monte Carlo Sampling Algorithm	395
11.4.2	Monte Carlo Rollouts with Function Approximators	396

5	Bootstrapping for Value Function Learning	398
11.5.1	Q-Learning	399
11.5.2	Deep Learning Models as Function Approximators	400
11.5.3	Example: Neural Network Specifics for Video Game Setting	403
11.5.4	On-Policy versus Off-Policy Methods: SARSA	404
11.5.5	Modeling States versus State-Action Pairs	405
6	Policy Gradient Methods	407
11.6.1	Finite Difference Methods	408
11.6.2	Likelihood Ratio Methods	409
11.6.3	Actor-Critic Methods	411
11.6.4	Continuous Action Spaces	413
7	Monte Carlo Tree Search	413
8	Case Studies	415
11.8.1	AlphaGo and AlphaZero for Go and Chess	415
11.8.2	Self-Learning Robots	420
11.8.2.1	Deep Learning of Locomotion Skills	420
11.8.2.2	Deep Learning of Visuomotor Skills	422
11.8.3	Building Conversational Systems: Deep Learning for Chatbots	423
11.8.4	Self-Driving Cars	425
11.8.5	Neural Architecture Search with Reinforcement Learning	428
9	Practical Challenges Associated with Safety	429
1.10	Summary	429
1.11	Bibliographic Notes and Software Resources	430
1.12	Exercises	432
	Advanced Topics in Deep Learning	435
2.1	Introduction	435
2.2	Attention Mechanisms	436
12.2.1	Recurrent Models of Visual Attention	437
12.2.2	Attention Mechanisms for Image Captioning	439
12.2.3	Soft Image Attention with Spatial Transformer	440
12.2.4	Attention Mechanisms for Machine Translation	442
12.2.5	Transformer Networks	446
12.2.5.1	How Self Attention Helps	446
12.2.5.2	The Self-Attention Module	447
12.2.5.3	Incorporating Positional Information	449
12.2.5.4	The Sequence-to-Sequence Transformer	450
12.2.5.5	Multihead Attention	450
12.2.6	Transformer-Based Pre-trained Language Models	451
12.2.6.1	GPT-n	452
12.2.6.2	BERT	454
12.2.6.3	T5	455
12.2.7	Vision Transformer (ViT)	457
12.2.8	Attention Mechanisms in Graphs	458
12.3	Neural Turing Machines	459
12.4	Adversarial Deep Learning	463
12.5	Generative Adversarial Networks (GANs)	467
12.5.1	Training a Generative Adversarial Network	468
12.5.2	Comparison with Variational Autoencoder	470

12.5.3	Using GANs for Generating Image Data	470
12.5.4	Conditional Generative Adversarial Networks	471
12.6	Competitive Learning	476
12.6.1	Vector Quantization	477
12.6.2	Kohonen Self-Organizing Map	478
12.7	Limitations of Neural Networks	480
12.7.1	An Aspirational Goal: Few Shot Learning	481
12.7.2	An Aspirational Goal: Energy-Efficient Learning	482
12.8	Summary	483
12.9	Bibliographic Notes and Software Resources	483
12.10	Exercises	485

Correction to: Neural Networks and Deep Learning **C1**

Bibliography **487**

Index **525**