Contents

Preface 1

Installation 9

Notation 13

1 Introduction 17
- 1.1 A Motivating Example . . . 18
- 1.2 The Key Components: Data, Models, and Algorithms . . . 20
- 1.3 Kinds of Machine Learning . . . 23
- 1.4 Roots . . . 35
- 1.5 The Road to Deep Learning . . . 37
- 1.6 Success Stories . . . 39

2 Preliminaries 43
- 2.1 Data Manipulation . . . 43
  - 2.1.1 Getting Started . . . 44
  - 2.1.2 Operations . . . 46
  - 2.1.3 Broadcasting Mechanism . . . 48
  - 2.1.4 Indexing and Slicing . . . 49
  - 2.1.5 Saving Memory . . . 49
  - 2.1.6 Conversion to Other Python Objects . . . 50
- 2.2 Data Preprocessing . . . 51
  - 2.2.1 Reading the Dataset . . . 51
  - 2.2.2 Handling Missing Data . . . 52
  - 2.2.3 Conversion to the `ndarray` Format . . . 53
- 2.3 Linear Algebra . . . 54
  - 2.3.1 Scalars . . . 54
  - 2.3.2 Vectors . . . 55
  - 2.3.3 Matrices . . . 56
  - 2.3.4 Tensors . . . 58
  - 2.3.5 Basic Properties of Tensor Arithmetic . . . 58
  - 2.3.6 Reduction . . . 59
  - 2.3.7 Dot Products . . . 61
  - 2.3.8 Matrix-Vector Products . . . 62
  - 2.3.9 Matrix-Matrix Multiplication . . . 63
  - 2.3.10 Norms . . . 64
  - 2.3.11 More on Linear Algebra . . . 65
- 2.4 Calculus . . . 67
  - 2.4.1 Derivatives and Differentiation . . . 67
  - 2.4.2 Partial Derivatives . . . 71

2.4.3 Gradients . . . 71
2.4.4 Chain Rule . . . 71
2.5 Automatic Differentiation . . . 72
2.5.1 A Simple Example . . . 73
2.5.2 Backward for Non-Scalar Variables . . . 74
2.5.3 Detaching Computation . . . 75
2.5.4 Computing the Gradient of Python Control Flow . . . 76
2.5.5 Training Mode and Prediction Mode . . . 77
2.6 Probability . . . 78
2.6.1 Basic Probability Theory . . . 79
2.6.2 Dealing with Multiple Random Variables . . . 82
2.6.3 Expectation and Variance . . . 85
2.7 Documentation . . . 86
2.7.1 Finding All the Functions and Classes in a Module . . . 87
2.7.2 Finding the Usage of Specific Functions and Classes . . . 87
2.7.3 API Documentation . . . 88

3 Linear Neural Networks **89**
3.1 Linear Regression . . . 89
3.1.1 Basic Elements of Linear Regression . . . 89
3.1.2 The Normal Distribution and Squared Loss . . . 95
3.1.3 From Linear Regression to Deep Networks . . . 97
3.2 Linear Regression Implementation from Scratch . . . 99
3.2.1 Generating the Dataset . . . 100
3.2.2 Reading the Dataset . . . 101
3.2.3 Initializing Model Parameters . . . 102
3.2.4 Defining the Model . . . 103
3.2.5 Defining the Loss Function . . . 103
3.2.6 Defining the Optimization Algorithm . . . 103
3.2.7 Training . . . 104
3.3 Concise Implementation of Linear Regression . . . 106
3.3.1 Generating the Dataset . . . 106
3.3.2 Reading the Dataset . . . 106
3.3.3 Defining the Model . . . 107
3.3.4 Initializing Model Parameters . . . 108
3.3.5 Defining the Loss Function . . . 108
3.3.6 Defining the Optimization Algorithm . . . 109
3.3.7 Training . . . 109
3.4 Softmax Regression . . . 110
3.4.1 Classification Problems . . . 111
3.4.2 Loss Function . . . 113
3.4.3 Information Theory Basics . . . 114
3.4.4 Model Prediction and Evaluation . . . 116
3.5 The Image Classification Dataset (Fashion-MNIST) . . . 117
3.5.1 Getting the Dataset . . . 117
3.5.2 Reading a Minibatch . . . 118
3.5.3 Putting All Things Together . . . 119
3.6 Implementation of Softmax Regression from Scratch . . . 120
3.6.1 Initializing Model Parameters . . . 121
3.6.2 The Softmax . . . 121
3.6.3 The Model . . . 122

3.6.4 The Loss Function . . . 123
3.6.5 Classification Accuracy . . . 123
3.6.6 Model Training . . . 125
3.6.7 Prediction . . . 127
3.7 Concise Implementation of Softmax Regression . . . 128
3.7.1 Initializing Model Parameters . . . 128
3.7.2 The Softmax . . . 128
3.7.3 Optimization Algorithm . . . 129
3.7.4 Training . . . 129

4 Multilayer Perceptrons **131**
4.1 Multilayer Perceptron . . . 131
4.1.1 Hidden Layers . . . 131
4.1.2 Activation Functions . . . 134
4.2 Implementation of Multilayer Perceptron from Scratch . . . 139
4.2.1 Initializing Model Parameters . . . 139
4.2.2 Activation Function . . . 140
4.2.3 The model . . . 140
4.2.4 The Loss Function . . . 140
4.2.5 Training . . . 140
4.3 Concise Implementation of Multilayer Perceptron . . . 142
4.3.1 The Model . . . 142
4.4 Model Selection, Underfitting and Overfitting . . . 143
4.4.1 Training Error and Generalization Error . . . 144
4.4.2 Model Selection . . . 146
4.4.3 Underfitting or Overfitting? . . . 147
4.4.4 Polynomial Regression . . . 149
4.5 Weight Decay . . . 153
4.5.1 Squared Norm Regularization . . . 154
4.5.2 High-Dimensional Linear Regression . . . 155
4.5.3 Implementation from Scratch . . . 156
4.5.4 Concise Implementation . . . 158
4.6 Dropout . . . 161
4.6.1 Overfitting Revisited . . . 161
4.6.2 Robustness through Perturbations . . . 161
4.6.3 Dropout in Practice . . . 162
4.6.4 Implementation from Scratch . . . 163
4.6.5 Concise Implementation . . . 165
4.7 Forward Propagation, Backward Propagation, and Computational Graphs . . . 167
4.7.1 Forward Propagation . . . 167
4.7.2 Computational Graph of Forward Propagation . . . 168
4.7.3 Backpropagation . . . 168
4.7.4 Training a Model . . . 170
4.8 Numerical Stability and Initialization . . . 171
4.8.1 Vanishing and Exploding Gradients . . . 171
4.8.2 Parameter Initialization . . . 173
4.9 Considering the Environment . . . 175
4.9.1 Distribution Shift . . . 176
4.9.2 A Taxonomy of Learning Problems . . . 182
4.9.3 Fairness, Accountability, and Transparency in Machine Learning . . . 183
4.10 Predicting House Prices on Kaggle . . . 184

4.10.1 Obtaining Data and Caching . . . 185
4.10.2 Kaggle . . . 186
4.10.3 Accessing and Reading the Dataset . . . 187
4.10.4 Data Preprocessing . . . 188
4.10.5 Training . . . 189
4.10.6 k-Fold Cross-Validation . . . 190
4.10.7 Model Selection . . . 191
4.10.8 Predict and Submit . . . 192

5 Deep Learning Computation **195**
5.1 Layers and Blocks . . . 195
5.1.1 A Custom Block . . . 198
5.1.2 The Sequential Block . . . 199
5.1.3 Blocks with Code . . . 200
5.1.4 Compilation . . . 201
5.2 Parameter Management . . . 202
5.2.1 Parameter Access . . . 203
5.2.2 Parameter Initialization . . . 207
5.2.3 Tied Parameters . . . 209
5.3 Deferred Initialization . . . 211
5.3.1 Instantiating a Network . . . 211
5.3.2 Deferred Initialization in Practice . . . 213
5.3.3 Forced Initialization . . . 213
5.4 Custom Layers . . . 215
5.4.1 Layers without Parameters . . . 215
5.4.2 Layers with Parameters . . . 216
5.5 File I/O . . . 218
5.5.1 Loading and Saving ndarrays . . . 218
5.5.2 Gluon Model Parameters . . . 219
5.6 GPUs . . . 220
5.6.1 Computing Devices . . . 222
5.6.2 ndarray and GPUs . . . 223
5.6.3 Gluon and GPUs . . . 225

6 Convolutional Neural Networks **227**
6.1 From Dense Layers to Convolutions . . . 228
6.1.1 Invariances . . . 228
6.1.2 Constraining the MLP . . . 229
6.1.3 Convolutions . . . 230
6.1.4 Waldo Revisited . . . 231
6.2 Convolutions for Images . . . 232
6.2.1 The Cross-Correlation Operator . . . 232
6.2.2 Convolutional Layers . . . 234
6.2.3 Object Edge Detection in Images . . . 234
6.2.4 Learning a Kernel . . . 235
6.2.5 Cross-Correlation and Convolution . . . 236
6.3 Padding and Stride . . . 237
6.3.1 Padding . . . 238
6.3.2 Stride . . . 240
6.4 Multiple Input and Output Channels . . . 241
6.4.1 Multiple Input Channels . . . 242

6.4.2 Multiple Output Channels . . . 243
6.4.3 1×1 Convolutional Layer . . . 244
6.5 Pooling . . . 246
6.5.1 Maximum Pooling and Average Pooling . . . 246
6.5.2 Padding and Stride . . . 248
6.5.3 Multiple Channels . . . 249
6.6 Convolutional Neural Networks (LeNet) . . . 250
6.6.1 LeNet . . . 251
6.6.2 Data Acquisition and Training . . . 253

7 Modern Convolutional Neural Networks 257
7.1 Deep Convolutional Neural Networks (AlexNet) . . . 257
7.1.1 Learning Feature Representation . . . 258
7.1.2 AlexNet . . . 261
7.1.3 Reading the Dataset . . . 264
7.1.4 Training . . . 264
7.2 Networks Using Blocks (VGG) . . . 265
7.2.1 VGG Blocks . . . 266
7.2.2 VGG Network . . . 266
7.2.3 Model Training . . . 268
7.3 Network in Network (NiN) . . . 269
7.3.1 NiN Blocks . . . 270
7.3.2 NiN Model . . . 271
7.3.3 Data Acquisition and Training . . . 272
7.4 Networks with Parallel Concatenations (GoogLeNet) . . . 273
7.4.1 Inception Blocks . . . 273
7.4.2 GoogLeNet Model . . . 275
7.4.3 Data Acquisition and Training . . . 277
7.5 Batch Normalization . . . 278
7.5.1 Training Deep Networks . . . 278
7.5.2 Batch Normalization Layers . . . 280
7.5.3 Implementation from Scratch . . . 281
7.5.4 Using a Batch Normalization LeNet . . . 282
7.5.5 Concise Implementation . . . 283
7.5.6 Controversy . . . 284
7.6 Residual Networks (ResNet) . . . 286
7.6.1 Function Classes . . . 286
7.6.2 Residual Blocks . . . 287
7.6.3 ResNet Model . . . 289
7.6.4 Data Acquisition and Training . . . 292
7.7 Densely Connected Networks (DenseNet) . . . 293
7.7.1 Function Decomposition . . . 293
7.7.2 Dense Blocks . . . 294
7.7.3 Transition Layers . . . 295
7.7.4 DenseNet Model . . . 296
7.7.5 Data Acquisition and Training . . . 296

8 Recurrent Neural Networks 299
8.1 Sequence Models . . . 299
8.1.1 Statistical Tools . . . 300
8.1.2 A Toy Example . . . 303

8.1.3 Predictions . . . 304
8.2 Text Preprocessing . . . 307
8.2.1 Reading the Dataset . . . 307
8.2.2 Tokenization . . . 308
8.2.3 Vocabulary . . . 308
8.2.4 Putting All Things Together . . . 310
8.3 Language Models and the Dataset . . . 311
8.3.1 Estimating a Language Model . . . 311
8.3.2 Markov Models and n-grams . . . 312
8.3.3 Natural Language Statistics . . . 313
8.3.4 Training Data Preparation . . . 315
8.4 Recurrent Neural Networks . . . 319
8.4.1 Recurrent Networks Without Hidden States . . . 319
8.4.2 Recurrent Networks with Hidden States . . . 320
8.4.3 Steps in a Language Model . . . 321
8.4.4 Perplexity . . . 322
8.5 Implementation of Recurrent Neural Networks from Scratch . . . 323
8.5.1 One-hot Encoding . . . 324
8.5.2 Initializing the Model Parameters . . . 324
8.5.3 RNN Model . . . 325
8.5.4 Prediction . . . 326
8.5.5 Gradient Clipping . . . 326
8.5.6 Training . . . 327
8.6 Concise Implementation of Recurrent Neural Networks . . . 331
8.6.1 Defining the Model . . . 331
8.6.2 Training and Predicting . . . 332
8.7 Backpropagation Through Time . . . 334
8.7.1 A Simplified Recurrent Network . . . 334
8.7.2 The Computational Graph . . . 336
8.7.3 BPTT in Detail . . . 337

9 Modern Recurrent Neural Networks 339
9.1 Gated Recurrent Units (GRU) . . . 339
9.1.1 Gating the Hidden State . . . 340
9.1.2 Implementation from Scratch . . . 342
9.1.3 Concise Implementation . . . 345
9.2 Long Short Term Memory (LSTM) . . . 346
9.2.1 Gated Memory Cells . . . 347
9.2.2 Implementation from Scratch . . . 350
9.2.3 Concise Implementation . . . 352
9.3 Deep Recurrent Neural Networks . . . 353
9.3.1 Functional Dependencies . . . 354
9.3.2 Concise Implementation . . . 355
9.3.3 Training . . . 355
9.4 Bidirectional Recurrent Neural Networks . . . 357
9.4.1 Dynamic Programming . . . 357
9.4.2 Bidirectional Model . . . 359
9.5 Machine Translation and the Dataset . . . 362
9.5.1 Reading and Preprocessing the Dataset . . . 363
9.5.2 Tokenization . . . 364
9.5.3 Vocabulary . . . 365

9.5.4 Loading the Dataset . . . 365
9.5.5 Putting All Things Together . . . 366
9.6 Encoder-Decoder Architecture . . . 367
9.6.1 Encoder . . . 367
9.6.2 Decoder . . . 367
9.6.3 Model . . . 368
9.7 Sequence to Sequence . . . 369
9.7.1 Encoder . . . 370
9.7.2 Decoder . . . 371
9.7.3 The Loss Function . . . 372
9.7.4 Training . . . 373
9.7.5 Predicting . . . 375
9.8 Beam Search . . . 376
9.8.1 Greedy Search . . . 376
9.8.2 Exhaustive Search . . . 378
9.8.3 Beam Search . . . 378

10 Attention Mechanisms **381**
10.1 Attention Mechanisms . . . 381
10.1.1 Dot Product Attention . . . 384
10.1.2 Multilayer Perceptron Attention . . . 385
10.2 Sequence to Sequence with Attention Mechanisms . . . 386
10.2.1 Decoder . . . 388
10.2.2 Training . . . 389
10.3 Transformer . . . 391
10.3.1 Multi-Head Attention . . . 392
10.3.2 Position-wise Feed-Forward Networks . . . 395
10.3.3 Add and Norm . . . 396
10.3.4 Positional Encoding . . . 397
10.3.5 Encoder . . . 398
10.3.6 Decoder . . . 399
10.3.7 Training . . . 401

11 Optimization Algorithms **405**
11.1 Optimization and Deep Learning . . . 405
11.1.1 Optimization and Estimation . . . 406
11.1.2 Optimization Challenges in Deep Learning . . . 407
11.2 Convexity . . . 411
11.2.1 Basics . . . 411
11.2.2 Properties . . . 414
11.2.3 Constraints . . . 417
11.3 Gradient Descent . . . 420
11.3.1 Gradient Descent in One Dimension . . . 420
11.3.2 Multivariate Gradient Descent . . . 423
11.3.3 Adaptive Methods . . . 425
11.4 Stochastic Gradient Descent . . . 430
11.4.1 Stochastic Gradient Updates . . . 430
11.4.2 Dynamic Learning Rate . . . 432
11.4.3 Convergence Analysis for Convex Objectives . . . 433
11.4.4 Stochastic Gradients and Finite Samples . . . 435
11.5 Minibatch Stochastic Gradient Descent . . . 436

11.5.1 Vectorization and Caches . . . 437
11.5.2 Minibatches . . . 439
11.5.3 Reading the Dataset . . . 440
11.5.4 Implementation from Scratch . . . 440
11.5.5 Concise Implementation . . . 444
11.6 Momentum . . . 445
11.6.1 Basics . . . 446
11.6.2 Practical Experiments . . . 450
11.6.3 Theoretical Analysis . . . 453
11.7 Adagrad . . . 455
11.7.1 Sparse Features and Learning Rates . . . 456
11.7.2 Preconditioning . . . 456
11.7.3 The Algorithm . . . 458
11.7.4 Implementation from Scratch . . . 460
11.7.5 Concise Implementation . . . 460
11.8 RMSProp . . . 462
11.8.1 The Algorithm . . . 462
11.8.2 Implementation from Scratch . . . 463
11.8.3 Concise Implementation . . . 465
11.9 Adadelta . . . 466
11.9.1 The Algorithm . . . 466
11.9.2 Implementation . . . 467
11.10 Adam . . . 469
11.10.1 The Algorithm . . . 469
11.10.2 Implementation . . . 470
11.10.3 Yogi . . . 471
11.11 Learning Rate Scheduling . . . 473
11.11.1 Toy Problem . . . 474
11.11.2 Schedulers . . . 475
11.11.3 Policies . . . 477

12 Computational Performance **483**
12.1 Compilers and Interpreters . . . 483
12.1.1 Symbolic Programming . . . 484
12.1.2 Hybrid Programming . . . 485
12.1.3 HybridSequential . . . 486
12.2 Asynchronous Computation . . . 490
12.2.1 Asynchrony via Backend . . . 491
12.2.2 Barriers and Blockers . . . 493
12.2.3 Improving Computation . . . 494
12.2.4 Improving Memory Footprint . . . 494
12.3 Automatic Parallelism . . . 497
12.3.1 Parallel Computation on CPUs and GPUs . . . 498
12.3.2 Parallel Computation and Communication . . . 499
12.4 Hardware . . . 501
12.4.1 Computers . . . 502
12.4.2 Memory . . . 503
12.4.3 Storage . . . 504
12.4.4 CPUs . . . 505
12.4.5 GPUs and other Accelerators . . . 508
12.4.6 Networks and Buses . . . 510

12.4.7 More Latency Numbers . . . 512
12.5 Training on Multiple GPUs . . . 514
12.5.1 Splitting the Problem . . . 514
12.5.2 Data Parallelism . . . 516
12.5.3 A Toy Network . . . 517
12.5.4 Data Synchronization . . . 518
12.5.5 Distributing Data . . . 519
12.5.6 Training . . . 520
12.5.7 Experiment . . . 521
12.6 Concise Implementation for Multiple GPUs . . . 522
12.6.1 A Toy Network . . . 523
12.6.2 Parameter Initialization and Logistics . . . 523
12.6.3 Training . . . 525
12.6.4 Experiments . . . 525
12.7 Parameter Servers . . . 527
12.7.1 Data Parallel Training . . . 527
12.7.2 Ring Synchronization . . . 530
12.7.3 Multi-Machine Training . . . 532
12.7.4 (key,value) Stores . . . 534

13 Computer Vision **537**
13.1 Image Augmentation . . . 537
13.1.1 Common Image Augmentation Method . . . 538
13.1.2 Using an Image Augmentation Training Model . . . 542
13.2 Fine Tuning . . . 545
13.2.1 Hot Dog Recognition . . . 546
13.3 Object Detection and Bounding Boxes . . . 551
13.3.1 Bounding Box . . . 552
13.4 Anchor Boxes . . . 553
13.4.1 Generating Multiple Anchor Boxes . . . 554
13.4.2 Intersection over Union . . . 556
13.4.3 Labeling Training Set Anchor Boxes . . . 556
13.4.4 Bounding Boxes for Prediction . . . 560
13.5 Multiscale Object Detection . . . 563
13.6 The Object Detection Dataset (Pikachu) . . . 566
13.6.1 Downloading the Dataset . . . 566
13.6.2 Reading the Dataset . . . 567
13.6.3 Demonstration . . . 568
13.7 Single Shot Multibox Detection (SSD) . . . 569
13.7.1 Model . . . 569
13.7.2 Training . . . 575
13.7.3 Prediction . . . 577
13.8 Region-based CNNs (R-CNNs) . . . 580
13.8.1 R-CNNs . . . 581
13.8.2 Fast R-CNN . . . 582
13.8.3 Faster R-CNN . . . 584
13.8.4 Mask R-CNN . . . 585
13.9 Semantic Segmentation and the Dataset . . . 586
13.9.1 Image Segmentation and Instance Segmentation . . . 586
13.9.2 The Pascal VOC2012 Semantic Segmentation Dataset . . . 587
13.10 Transposed Convolution . . . 592

13.10.1 Basic 2D Transposed Convolution . . . 592
13.10.2 Padding, Strides, and Channels . . . 593
13.10.3 Analogy to Matrix Transposition . . . 594
13.11 Fully Convolutional Networks (FCN) . . . 596
13.11.1 Constructing a Model . . . 596
13.11.2 Initializing the Transposed Convolution Layer . . . 598
13.11.3 Reading the Dataset . . . 600
13.11.4 Training . . . 600
13.11.5 Prediction . . . 601
13.12 Neural Style Transfer . . . 603
13.12.1 Technique . . . 603
13.12.2 Reading the Content and Style Images . . . 604
13.12.3 Preprocessing and Postprocessing . . . 605
13.12.4 Extracting Features . . . 606
13.12.5 Defining the Loss Function . . . 607
13.12.6 Creating and Initializing the Composite Image . . . 608
13.12.7 Training . . . 609
13.13 Image Classification (CIFAR-10) on Kaggle . . . 612
13.13.1 Obtaining and Organizing the Dataset . . . 613
13.13.2 Image Augmentation . . . 616
13.13.3 Reading the Dataset . . . 616
13.13.4 Defining the Model . . . 617
13.13.5 Defining the Training Functions . . . 618
13.13.6 Training and Validating the Model . . . 618
13.13.7 Classifying the Testing Set and Submitting Results on Kaggle . . . 619
13.14 Dog Breed Identification (ImageNet Dogs) on Kaggle . . . 620
13.14.1 Obtaining and Organizing the Dataset . . . 621
13.14.2 Image Augmentation . . . 622
13.14.3 Reading the Dataset . . . 623
13.14.4 Defining the Model . . . 623
13.14.5 Defining the Training Functions . . . 624
13.14.6 Training and Validating the Model . . . 625
13.14.7 Classifying the Testing Set and Submit Results on Kaggle . . . 625

14 Natural Language Processing **627**
14.1 Word Embedding (word2vec) . . . 627
14.1.1 Why Not Use One-hot Vectors? . . . 627
14.1.2 The Skip-Gram Model . . . 628
14.1.3 The Continuous Bag of Words (CBOW) Model . . . 630
14.2 Approximate Training for Word2vec . . . 632
14.2.1 Negative Sampling . . . 632
14.2.2 Hierarchical Softmax . . . 633
14.3 The Dataset for Word2vec . . . 635
14.3.1 Reading and Preprocessing the Dataset . . . 635
14.3.2 Subsampling . . . 636
14.3.3 Loading the Dataset . . . 638
14.3.4 Putting All Things Together . . . 641
14.4 Implementation of Word2vec . . . 642
14.4.1 The Skip-Gram Model . . . 643
14.4.2 Training . . . 644
14.4.3 Applying the Word Embedding Model . . . 646

14.5 Subword Embedding (fastText) . . . 647
14.6 Word Embedding with Global Vectors (GloVe) . . . 648
14.6.1 The GloVe Model . . . 649
14.6.2 Understanding GloVe from Conditional Probability Ratios . . . 650
14.7 Finding Synonyms and Analogies . . . 651
14.7.1 Using Pre-Trained Word Vectors . . . 652
14.7.2 Applying Pre-Trained Word Vectors . . . 653
14.8 Text Classification and the Dataset . . . 655
14.8.1 The Text Sentiment Classification Dataset . . . 655
14.8.2 Putting All Things Together . . . 658
14.9 Text Sentiment Classification: Using Recurrent Neural Networks . . . 658
14.9.1 Using a Recurrent Neural Network Model . . . 659
14.10 Text Sentiment Classification: Using Convolutional Neural Networks (textCNN) . . . 662
14.10.1 One-Dimensional Convolutional Layer . . . 663
14.10.2 Max-Over-Time Pooling Layer . . . 665
14.10.3 The TextCNN Model . . . 665

15 Recommender Systems **671**
15.1 Overview of Recommender Systems . . . 671
15.1.1 Collaborative Filtering . . . 672
15.1.2 Explicit Feedback and Implicit Feedback . . . 673
15.1.3 Recommendation Tasks . . . 673
15.2 The MovieLens Dataset . . . 674
15.2.1 Getting the Data . . . 674
15.2.2 Statistics of the Dataset . . . 675
15.2.3 Splitting the dataset . . . 676
15.2.4 Loading the data . . . 677
15.3 Matrix Factorization . . . 678
15.3.1 The Matrix Factorization Model . . . 679
15.3.2 Model Implementation . . . 680
15.3.3 Evaluation Measures . . . 680
15.3.4 Training and Evaluating the Model . . . 681
15.4 AutoRec: Rating Prediction with Autoencoders . . . 683
15.4.1 Model . . . 683
15.4.2 Implementing the Model . . . 684
15.4.3 Reimplementing the Evaluator . . . 684
15.4.4 Training and Evaluating the Model . . . 685
15.5 Personalized Ranking for Recommender Systems . . . 686
15.5.1 Bayesian Personalized Ranking Loss and its Implementation . . . 687
15.5.2 Hinge Loss and its Implementation . . . 688
15.6 Neural Collaborative Filtering for Personalized Ranking . . . 689
15.6.1 The NeuMF model . . . 690
15.6.2 Model Implementation . . . 691
15.6.3 Customized Dataset with Negative Sampling . . . 692
15.6.4 Evaluator . . . 692
15.6.5 Training and Evaluating the Model . . . 694
15.7 Sequence-Aware Recommender Systems . . . 696
15.7.1 Model Architectures . . . 696
15.7.2 Model Implementation . . . 698
15.7.3 Sequential Dataset with Negative Sampling . . . 699
15.7.4 Load the MovieLens 100K dataset . . . 700

15.7.5 Train the Model . . . 701
15.8 Feature-Rich Recommender Systems . . . 702
15.8.1 An Online Advertising Dataset . . . 703
15.8.2 Dataset Wrapper . . . 703
15.9 Factorization Machines . . . 705
15.9.1 2-Way Factorization Machines . . . 705
15.9.2 An Efficient Optimization Criterion . . . 706
15.9.3 Model Implementation . . . 707
15.9.4 Load the Advertising Dataset . . . 707
15.9.5 Train the Model . . . 707
15.10 Deep Factorization Machines . . . 709
15.10.1 Model Architectures . . . 709
15.10.2 Implemenation of DeepFM . . . 710
15.10.3 Training and Evaluating the Model . . . 711

16 Generative Adversarial Networks 713
16.1 Generative Adversarial Networks . . . 713
16.1.1 Generate some "real" data . . . 715
16.1.2 Generator . . . 716
16.1.3 Discriminator . . . 716
16.1.4 Training . . . 716
16.2 Deep Convolutional Generative Adversarial Networks . . . 719
16.2.1 The Pokemon Dataset . . . 719
16.2.2 The Generator . . . 720
16.2.3 Discriminator . . . 722
16.2.4 Training . . . 723

17 Appendix: Mathematics for Deep Learning 727
17.1 Geometry and Linear Algebraic Operations . . . 728
17.1.1 Geometry of Vectors . . . 728
17.1.2 Dot Products and Angles . . . 730
17.1.3 Hyperplanes . . . 732
17.1.4 Geometry of Linear Transformations . . . 735
17.1.5 Linear Dependence . . . 737
17.1.6 Rank . . . 737
17.1.7 Invertibility . . . 738
17.1.8 Determinant . . . 739
17.1.9 Tensors and Common Linear Algebra Operations . . . 740
17.2 Eigendecompositions . . . 744
17.2.1 Finding Eigenvalues . . . 744
17.2.2 Decomposing Matrices . . . 745
17.2.3 Operations on Eigendecompositions . . . 746
17.2.4 Eigendecompositions of Symmetric Matrices . . . 746
17.2.5 Gershgorin Circle Theorem . . . 747
17.2.6 A Useful Application: The Growth of Iterated Maps . . . 748
17.2.7 Conclusions . . . 752
17.3 Single Variable Calculus . . . 753
17.3.1 Differential Calculus . . . 753
17.3.2 Rules of Calculus . . . 756
17.4 Multivariable Calculus . . . 763
17.4.1 Higher-Dimensional Differentiation . . . 764

17.4.2 Geometry of Gradients and Gradient Descent . . . 765
17.4.3 A Note on Mathematical Optimization . . . 766
17.4.4 Multivariate Chain Rule . . . 767
17.4.5 The Backpropagation Algorithm . . . 769
17.4.6 Hessians . . . 772
17.4.7 A Little Matrix Calculus . . . 774
17.5 Integral Calculus . . . 779
17.5.1 Geometric Interpretation . . . 779
17.5.2 The Fundamental Theorem of Calculus . . . 781
17.5.3 Change of Variables . . . 783
17.5.4 A Comment on Sign Conventions . . . 784
17.5.5 Multiple Integrals . . . 785
17.5.6 Change of Variables in Multiple Integrals . . . 787
17.6 Random Variables . . . 788
17.6.1 Continuous Random Variables . . . 789
17.7 Maximum Likelihood . . . 806
17.7.1 The Maximum Likelihood Principle . . . 806
17.7.2 Numerical Optimization and the Negative Log-Likelihood . . . 808
17.7.3 Maximum Likelihood for Continuous Variables . . . 809
17.8 Naive Bayes . . . 811
17.8.1 Optical Character Recognition . . . 811
17.8.2 The Probabilistic Model for Classification . . . 813
17.8.3 The Naive Bayes Classifier . . . 813
17.8.4 Training . . . 814
17.9 Statistics . . . 818
17.9.1 Evaluating and Comparing Estimators . . . 818
17.9.2 Conducting Hypothesis Tests . . . 822
17.9.3 Constructing Confidence Intervals . . . 826
17.10 Information Theory . . . 829
17.10.1 Information . . . 829
17.10.2 Entropy . . . 831
17.10.3 Mutual Information . . . 833
17.10.4 Kullback–Leibler Divergence . . . 837
17.10.5 Cross Entropy . . . 839

18 Appendix: Tools for Deep Learning **843**
18.1 Using Jupyter . . . 843
18.1.1 Editing and Running the Code Locally . . . 843
18.1.2 Advanced Options . . . 847
18.2 Using Amazon SageMaker . . . 848
18.2.1 Registering Account and Logging In . . . 848
18.2.2 Creating an SageMaker Instance . . . 849
18.2.3 Running and Stopping an Instance . . . 850
18.2.4 Updating Notebooks . . . 852
18.3 Using AWS EC2 Instances . . . 852
18.3.1 Creating and Running an EC2 Instance . . . 853
18.3.2 Installing CUDA . . . 857
18.3.3 Installing MXNet and Downloading the D2L Notebooks . . . 858
18.3.4 Running Jupyter . . . 860
18.3.5 Closing Unused Instances . . . 860
18.4 Using Google Colab . . . 861

18.5 Selecting Servers and GPUs . . . 862
18.5.1 Selecting Servers . . . 862
18.5.2 Selecting GPUs . . . 864
18.6 Contributing to This Book . . . 867
18.6.1 From Reader to Contributor in 6 Steps . . . 867
18.7 d2l API Document . . . 871

Bibliography **877**