Artificial Intelligence: A Textbook

Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown Heights, New York

March 23, 2021

To my wife Lata, my daughter Sayani,
and all my computer science instructors

Contents

1 An Introduction to Artificial Intelligence **1**
1.1 Introduction . . . 1
1.2 The Two Schools of Thought . . . 2
1.2.1 Induction and Deduction: A Historical View . . . 8
1.3 Artificial General Intelligence . . . 12
1.3.1 The Turing Test . . . 12
1.4 The Concept of Agent . . . 13
1.4.1 Types of Environments . . . 15
1.5 Deductive Reasoning in Artificial Intelligence . . . 16
1.5.1 Examples of Deductive Reasoning in Artificial Intelligence . . . 16
1.5.1.1 Constraint Satisfaction Problem . . . 16
1.5.1.2 Solving NP-Hard Problems . . . 18
1.5.1.3 Game Playing . . . 19
1.5.1.4 Planning . . . 20
1.5.1.5 Expert Systems . . . 21
1.5.2 Classical Methods for Deductive Reasoning . . . 22
1.5.2.1 Search-Based Methods . . . 22
1.5.2.2 Logic Programming . . . 23
1.5.3 Strengths and Limitations of Deductive Reasoning . . . 24
1.6 Inductive Learning in Artificial Intelligence . . . 24
1.6.1 Types of Learning . . . 26
1.6.2 Unsupervised Learning Tasks . . . 27
1.6.3 Supervised Learning Tasks . . . 29
1.7 Biological Evolution in Artificial Intelligence . . . 31
1.8 Summary . . . 32
1.9 Further Reading . . . 33
1.10 Exercises . . . 33

2 Searching State Spaces **35**
2.1 Introduction . . . 35
2.1.1 State Space as a Graph . . . 37
2.2 Uninformed Search Algorithms . . . 39
2.2.1 Case Study: Eight-Puzzle Problem . . . 45

2.2.2 Case Study: Online Maze Search . . . 46
2.2.3 Improving Efficiency with Bidirectional Search . . . 47
2.3 Informed Search: Best-First Search . . . 49
2.3.1 Greedy Best-First Search . . . 51
2.3.2 A^*-Search . . . 52
2.4 Local Search with State-Specific Loss Functions . . . 53
2.4.1 Hill Climbing . . . 55
2.4.1.1 The Problem of Local Optima . . . 57
2.4.2 Tabu Search . . . 58
2.4.3 Simulated Annealing . . . 60
2.5 Genetic Algorithms . . . 62
2.6 The Constraint Satisfaction Problem . . . 64
2.6.1 Traveling Salesperson Problem as Constraint Satisfaction . . . 65
2.6.2 Graph Coloring as Constraint Satisfaction . . . 65
2.6.3 Sudoku as Constraint Satisfaction . . . 66
2.6.4 Search Algorithms for Constraint Satisfaction . . . 67
2.6.5 Leveraging State-Specific Loss Values . . . 68
2.7 Summary . . . 68
2.8 Further Reading . . . 69
2.9 Exercises . . . 69

3 Multiagent Search 71
3.1 Introduction . . . 71
3.2 Uninformed Search: AND-OR Search Trees . . . 73
3.2.1 Handling More than Two Agents . . . 76
3.2.2 Handling Non-Deterministic Environments . . . 77
3.3 Informed Search Trees with State-Specific Loss Functions . . . 77
3.3.1 Heuristic Variations . . . 81
3.3.2 Adaptation to Adversarial Environments . . . 82
3.3.3 Prestoring Subtrees . . . 84
3.3.4 Challenges in Designing Evaluation Functions . . . 85
3.3.5 Weaknesses of Minimax Trees . . . 87
3.4 Alpha-Beta Pruning . . . 88
3.4.1 Importance of Branch Evaluation Order . . . 91
3.5 Monte Carlo Tree Search: The Inductive View . . . 92
3.5.1 Enhancements to the Expected Outcome Model . . . 94
3.5.2 Deductive versus Inductive: Minimax and Monte Carlo Trees . . . 99
3.5.3 Application to Non-Deterministic and Partially Observable Games . 100
3.6 Summary . . . 101
3.7 Further Reading . . . 102
3.8 Exercises . . . 102

4 Propositional Logic 105
4.1 Introduction . . . 105
4.2 Propositional Logic: The Basics . . . 107
4.2.1 Truth Tables . . . 111
4.3 Laws of Propositional Logic . . . 112
4.3.1 Useful Properties of Implication and Equivalence . . . 114
4.3.2 Tautologies and Satisfiability . . . 115

4.3.3 Clauses and Canonical Forms 116
4.4 Propositional Logic as a Precursor to Expert Systems 118
4.5 Equivalence of Expressions in Propositional Logic 120
4.6 The Basics of Proofs in Knowledge Bases 122
4.7 The Method of Proof by Contradiction 124
4.8 Efficient Entailment with Definite Clauses 129
4.8.1 Forward Chaining 129
4.8.2 Backward Chaining 131
4.8.3 Comparing Forward and Backward Chaining 132
4.9 Summary 133
4.10 Further Reading 133
4.11 Exercises 133

5 First-Order Logic 137
5.1 Introduction 137
5.2 The Basics of First-Order Logic 139
5.2.1 The Use of Quantifiers 141
5.2.2 Functions in First-Order Logic 145
5.2.3 How First-Order Logic Builds on Propositional Logic 146
5.2.4 Standardization Issues and Scope Extension 149
5.2.5 Interaction of Negation with Quantifiers 150
5.2.6 Substitution and Skolemization 151
5.2.7 Why First-Order Logic Is More Expressive 154
5.3 Populating a Knowledge Base 155
5.4 Example of Expert System with First-Order Logic 157
5.5 Systematic Inferencing Procedures 159
5.5.1 The Method of Proof by Contradiction 159
5.5.2 Forward Chaining 162
5.5.3 Backward Chaining 163
5.6 Summary 163
5.7 Further Reading 163
5.8 Exercises 164

6 Machine Learning: The Inductive View 167
6.1 Introduction 167
6.2 Linear Regression 170
6.2.1 Stochastic Gradient Descent 171
6.2.2 Matrix-Based Solution 172
6.2.3 Use of Bias 173
6.2.4 Why Is Regularization Important? 174
6.3 Least-Squares Classification 175
6.3.1 Problems with Least-Squares Classification 178
6.4 The Support Vector Machine 179
6.4.1 Mini-Batch Stochastic Gradient Descent 180
6.5 Logistic Regression 180
6.5.1 Computing Gradients 181
6.5.2 Comparing the SVM and Logistic Regression 181
6.5.3 Logistic Regression as a Probabilistic Classifier 182
6.6 Multiclass Setting 184

6.6.1 One-Against-Rest and One-Against-One Voting 184
6.6.2 Multinomial Logistic Regression 185
6.6.2.1 Stochastic Gradient Descent 185
6.7 The Naïve Bayes Model 186
6.8 Nearest Neighbor Classifier 189
6.9 Decision Trees 190
6.9.1 Training Phase of Decision Tree Construction 191
6.9.2 Splitting a Node 194
6.9.3 Generalizing Decision Trees to Random Forests 196
6.10 Rule-Based Classifiers 197
6.10.1 Sequential Covering Algorithms 198
6.10.1.1 Learn-One-Rule 199
6.10.2 Comparing Rule-Based Classifiers to Logical Rules in Expert Systems 200
6.11 Evaluation of Classification 200
6.11.1 Segmenting into Training and Testing Portions 201
6.11.1.1 Hold-Out 202
6.11.1.2 Cross-Validation 202
6.11.2 Absolute Accuracy Measures 203
6.11.2.1 Accuracy of Classification 203
6.11.2.2 Accuracy of Regression 204
6.11.3 Ranking Measures 205
6.11.3.1 Receiver Operating Characteristic 205
6.12 Summary 209
6.13 Further Reading 209
6.14 Exercises 209

7 Neural Networks 211
7.1 Introduction 211
7.2 An Introduction to Computational Graphs 213
7.2.1 Neural Networks as Directed Computational Graphs 216
7.2.2 Softmax Activation Function 218
7.2.3 Common Loss Functions 219
7.2.4 How Nonlinearity Increases Expressive Power 219
7.3 Optimization in Directed Acyclic Graphs 221
7.3.1 The Challenge of Computational Graphs 221
7.3.2 The Broad Framework for Gradient Computation 223
7.3.3 Computing Node-to-Node Derivatives Using Brute Force 224
7.3.4 Dynamic Programming for Computing Node-to-Node Derivatives . . 227
7.3.4.1 Example of Computing Node-to-Node Derivatives 229
7.3.5 Converting Node-to-Node Derivatives into Loss-to-Weight Derivatives 232
7.3.5.1 Example of Computing Loss-to-Weight Derivatives 233
7.3.6 Computational Graphs with Vector Variables 234
7.4 Application: Backpropagation in Neural Networks 237
7.4.1 Derivatives of Common Activation Functions 239
7.4.2 The Special Case of Softmax 240
7.4.3 Vector-Centric Backpropagation 240
7.4.4 Example of Vector-Centric Backpropagation 243
7.5 A General View of Computational Graphs 245
7.6 Summary 247

7.7 Further Reading 248
7.8 Exercises 248

8 Domain-Specific Neural Architectures 253
8.1 Introduction 253
8.2 Principles Underlying Convolutional Neural Networks 254
8.3 The Basic Structure of a Convolutional Network 256
8.3.1 Padding 261
8.3.2 Strides 262
8.3.3 Typical Settings 263
8.3.4 The ReLU Layer 264
8.3.5 Pooling 264
8.3.6 Fully Connected Layers 266
8.3.7 The Interleaving between Layers 267
8.3.8 Hierarchical Feature Engineering 268
8.4 Case Studies of Convolutional Architectures 270
8.4.1 AlexNet 271
8.4.2 VGG 273
8.4.3 ResNet 276
8.5 Principles Underlying Recurrent Neural Networks 278
8.6 The Architecture of Recurrent Neural Networks 280
8.6.1 Language Modeling Example of RNN 283
8.6.1.1 Generating a Language Sample 284
8.6.2 Backpropagation Through Time 286
8.6.3 Multilayer Recurrent Networks 288
8.7 Long Short-Term Memory (LSTM) 290
8.8 Applications of Domain-Specific Architectures 293
8.8.1 Application to Automatic Image Captioning 294
8.8.2 Sequence-to-Sequence Learning and Machine Translation 295
8.9 Summary 297
8.10 Further Reading 297
8.11 Exercises 297

9 Unsupervised Learning 299
9.1 Introduction 299
9.2 Dimensionality Reduction and Matrix Factorization 300
9.2.1 Symmetric Matrix Factorization 301
9.2.2 Singular Value Decomposition 302
9.2.2.1 Example of SVD 305
9.2.2.2 Alternate Optima via Gradient Descent 306
9.2.3 Nonnegative Matrix Factorization 308
9.2.3.1 Interpreting Nonnegative Matrix Factorization 310
9.2.4 Dimensionality Reduction with Neural Networks 313
9.2.4.1 Linear Autoencoder with a Single Hidden Layer 314
9.2.4.2 Nonlinear Activations 316
9.3 Clustering 318
9.3.1 Representative-based Algorithms 318
9.3.2 Bottom-up Agglomerative Methods 320
9.3.2.1 Group-based Statistics 322

9.3.3 Top-down Divisive Methods 325
9.3.3.1 Bisecting k-Means 325
9.3.4 Probabilistic Model-based Algorithms 325
9.3.5 Kohonen Self-Organizing Map 329
9.3.6 Combining Row and Column Similarities: Spectral Clustering 331
9.4 Why Unsupervised Learning Is Important 333
9.4.1 Feature Engineering for Machine Learning 333
9.4.2 Radial Basis Function Networks for Feature Engineering 335
9.4.3 Semisupervised Learning 337
9.4.3.1 Self-Training 337
9.4.3.2 Co-Training 337
9.4.3.3 Unsupervised Pretraining in Multilayer Neural Networks . 338
9.5 Summary 341
9.6 Further Reading 341
9.7 Exercises 341

10 Reinforcement Learning 343

10.1 Introduction 343
10.2 Stateless Algorithms: Multi-Armed Bandits 345
10.2.1 Naïve Algorithm 345
10.2.2 ϵ-Greedy Algorithm 346
10.2.3 Upper Bounding Methods 346
10.3 Reinforcement Learning Framework 347
10.4 Monte Carlo Sampling 349
10.4.1 Monte Carlo Sampling Algorithm 350
10.4.2 Monte Carlo Rollouts with Function Approximators 351
10.4.3 Connections to Monte Carlo Tree Search 353
10.5 Bootstrapping and Temporal Difference Learning 354
10.5.1 Q-Learning 355
10.5.2 Using Function Approximators 356
10.5.3 Example: Neural Network Specifics for Video Game Setting 359
10.5.4 On-Policy versus Off-Policy Methods: SARSA 360
10.5.5 Modeling States versus State-Action Pairs 361
10.6 Policy Gradient Methods 363
10.6.1 The Likelihood Ratio Principle 365
10.6.2 Combining Supervised Learning with Policy Gradients 366
10.6.3 Actor-Critic Methods 367
10.6.4 Continuous Action Spaces 368
10.6.5 Advantages and Disadvantages of Policy Gradients 369
10.7 Revisiting Monte Carlo Tree Search 369
10.8 Case Studies 371
10.8.1 AlphaGo: Championship Level Play at Go 371
10.8.1.1 AlphaZero: Enhancements to Zero Human Knowledge . . . 374
10.8.2 Self-Learning Robots 376
10.8.2.1 Deep Learning of Locomotion Skills 376
10.8.2.2 Deep Learning of Visuomotor Skills 377
10.8.3 Self-Driving Cars 379
10.9 Weaknesses of Reinforcement Learning 381
10.10Summary 382

10.11Further Reading 383
10.12Exercises 383

11 Probabilistic Graphical Models 385
11.1 Introduction 385
11.2 Bayesian Networks 386
11.3 Rudimentary Probabilistic Models in Machine Learning 390
11.4 The Boltzmann Machine 392
11.4.1 How a Boltzmann Machine Generates Data 394
11.4.2 Learning the Weights of a Boltzmann Machine 395
11.5 Restricted Boltzmann Machines 396
11.5.1 Training the RBM 399
11.5.2 Contrastive Divergence Algorithm 399
11.5.3 Practical Issues and Improvisations 400
11.6 Applications of Restricted Boltzmann Machines 401
11.6.1 Dimensionality Reduction and Data Reconstruction 402
11.6.2 RBMs for Collaborative Filtering 404
11.7 Summary 407
11.8 Further Reading 407
11.9 Exercises 408

12 Knowledge Graphs 409
12.1 Introduction 409
12.2 An Overview of Knowledge Graphs 413
12.2.1 Example: WordNet 419
12.2.2 Example: YAGO 420
12.2.3 Example: DBpedia 420
12.2.4 Example: Freebase 421
12.2.5 Example: Wikidata 422
12.2.6 Example: Gene Ontology 422
12.3 How to Construct a Knowledge Graph 423
12.3.1 First-Order Logic to Knowledge Graphs 425
12.3.2 Extraction from Unstructured Data 425
12.3.3 Handling Incompleteness 426
12.4 Applications of Knowledge Graphs 428
12.4.1 Knowledge Graphs in Search 428
12.4.2 Clustering Knowledge Graphs 430
12.4.3 Entity Classification 431
12.4.4 Link Prediction and Relationship Classification 432
12.4.5 Recommender Systems 433
12.5 Summary 434
12.6 Further Reading 434
12.7 Exercises 435

13 Integrating Reasoning and Learning 437
13.1 Introduction 437
13.2 The Bias-Variance Trade-Off 442
13.2.1 Formal View 443
13.3 A Generic Deductive-Inductive Ensemble 447

13.3.1 Inductive Ensemble Methods . . . 447
13.4 Transfer Learning . . . 450
13.4.1 Image Data . . . 452
13.4.2 Text Data . . . 453
13.4.3 Cross-Domain Transfer Learning . . . 453
13.5 Lifelong Machine Learning . . . 457
13.5.1 An Instructive Example of Lifelong Learning . . . 457
13.6 Neuro-Symbolic Artificial Intelligence . . . 461
13.6.1 Question Answering on Images . . . 463
13.7 Summary . . . 465
13.8 Further Reading . . . 465
13.9 Exercises . . . 466

Preface

> "AI is likely to be either the best or the worst thing that happened to humanity."– Stephen Hawking

Artificial intelligence is a field that spans work from multiple communities, including classical logic programming, machine learning, and data mining. Since the founding of the field, there has a clear dichotomy between the *deductive reasoning* and the *inductive learning* forms of artificial intelligence. In the deductive reasoning view, one starts with various forms of domain knowledge (which are often stored as knowledge bases), and these forms of domain knowledge are used in order to make inferences. Such methods are often highly interpretable. The domain knowledge can be used in order to create hypotheses, which are then leveraged to make predictions. For example, in a chess game, the domain knowledge about the importance and position of pieces can be used to create a hypothesis about the quality of a position. This hypothesis can be used to predict moves by searching a tree of possible moves up to a specific number of moves. In learning methods, data-driven evidence is used to *learn* how to make predictions. For example, it is possible to generate data from chess games using self play, and then learn which moves are best for any particular (type of) position. Since the number of possible alternative move sequences in chess is too large to evaluate explicitly, chess programs often use various types of machine learning methods to relate typical patterns of pieces on the board to make predictions from carefully selected sequences. This approach is somewhat similar to how humans make chess moves. In the early years, deductive reasoning methods were more popular, although inductive learning methods have become increasingly popular in recent years. Many books in artificial intelligence tend to focus predominantly on deductive reasoning as a legacy from its dominance during the early years. This book has attempted to strike a balance between deductive reasoning and inductive learning.

The main disadvantage of inductive learning methods is that they are not interpretable, and they often require a lot of data. A key point is that humans do not require a lot of data to learn. For example, a child is often able to learn to recognize a truck with the use of a small number of examples. Although the best solutions to many problems in artificial intelligence integrate methods from both these areas, there is often little discussion of this type of integration. This textbook focuses on giving an integrated view of artificial intelligence, along with a discussion of the advantages of different views of artificial intelligence.

After presenting a broad overview in Chapter 1, the remaining portions of this book primarily belong to three categories:

1. *Methods based on deductive reasoning:* Chapters 2 through 5 discuss deductive reasoning methods. The primary focus areas include search and logic.

2. *Methods based on inductive learning:* Learning methods are discussed in Chapters 6 to 10. The topics covered include classification, neural networks, unsupervised learning, probabilistic graphical models, and reinforcement learning.

3. *Methods based on both reasoning and learning:* Chapter 11 to 13 discuss a number of methods that have aspects of both reasoning and learning. This include techniques like Bayesian networks, knowledge graphs, and neuro-symbolic artificial intelligence.

A number of topics of recent importance, such as transfer learning and lifelong learning, are also discussed in this book.

Throughout this book, a vector or a multidimensional data point is annotated with a bar, such as $\overline{X}$ or $\overline{y}$. A vector or multidimensional point may be denoted by either small letters or capital letters, as long as it has a bar. Vector dot products are denoted by centered dots, such as $\overline{X} \cdot \overline{Y}$. A matrix is denoted in capital letters without a bar, such as R. Throughout the book, the $n \times d$ matrix corresponding to the entire training data set is denoted by D, with n data points and d dimensions. The individual data points in D are therefore d-dimensional row vectors, and are often denoted by $\overline{X}_1 \ldots \overline{X}_n$. On the other hand, vectors with one component for each data point are usually n-dimensional column vectors. An example is the n-dimensional column vector $\overline{y}$ of class variables of n data points. An observed value y_i is distinguished from a predicted value $\hat{y}_i$ by a circumflex at the top of the variable.