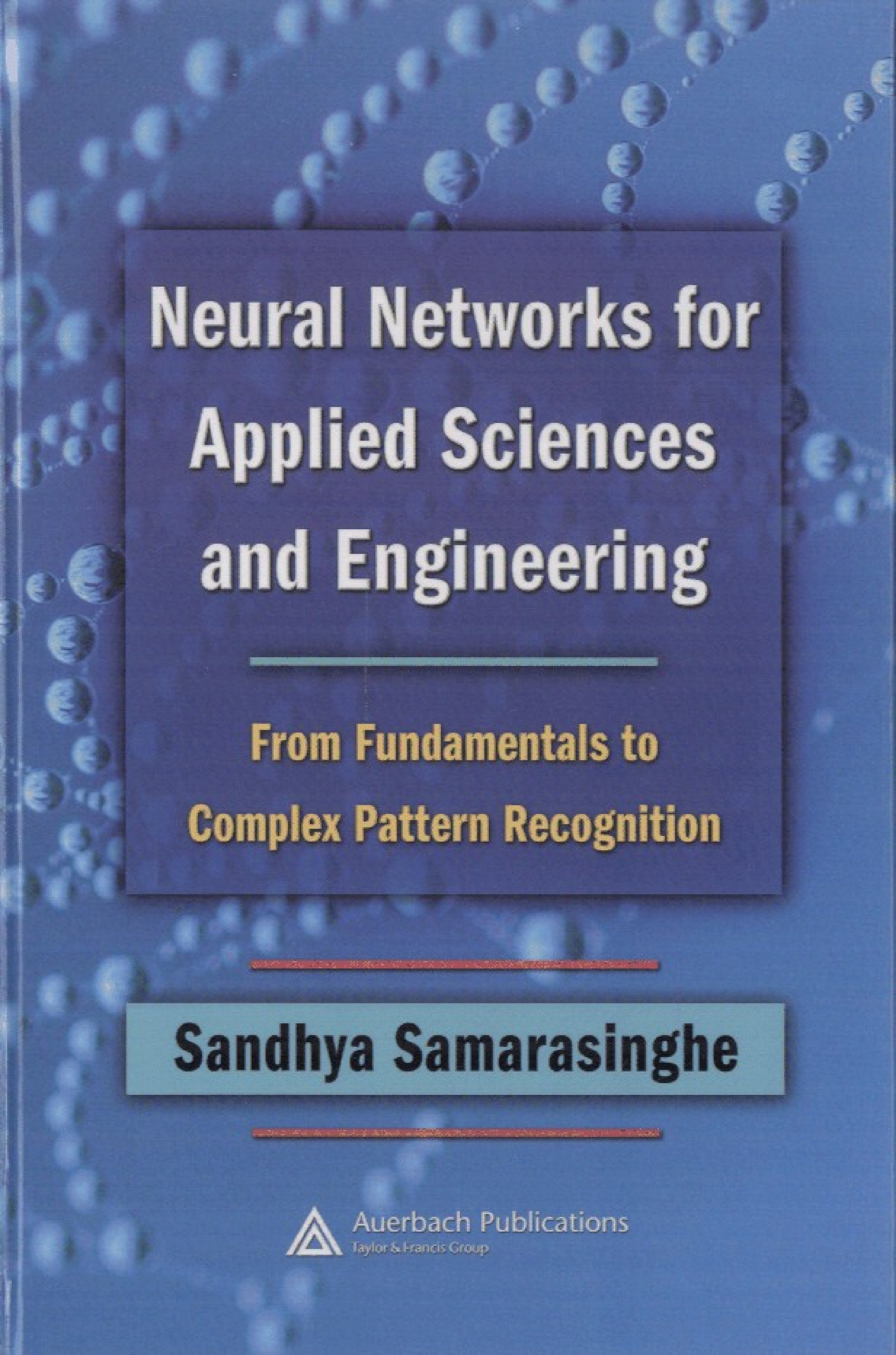

Neural Networks for Applied Sciences and Engineering

From Fundamentals to Complex Pattern Recognition

Sandhya Samarasinghe

Auerbach Publications
Taylor & Francis Group

Contents

Preface xvii
Acknowledgments xxi
About the Author xxiii

1 From Data to Models: Complexity and Challenges in Understanding Biological, Ecological, and Natural Systems 1
1.1: Introduction 1
1.2: Layout of the Book 4
References 7

2 Fundamentals of Neural Networks and Models for Linear Data Analysis 11
2.1: Introduction and Overview 11
2.2: Neural Networks and Their Capabilities 12
2.3: Inspirations from Biology 16
2.4: Modeling Information Processing in Neurons 18
2.5: Neuron Models and Learning Strategies 19
2.5.1: Threshold Neuron as a Simple Classifier 20
2.5.2: Learning Models for Neurons and Neural Assemblies 23
2.5.2.1: Hebbian Learning 23
2.5.2.2: Unsupervised or Competitive Learning 26
2.5.2.3: Supervised Learning 26
2.5.3: Perceptron with Supervised Learning as a Classifier 27
2.5.3.1: Perceptron Learning Algorithm 28
2.5.3.2: A Practical Example of Perceptron on a Larger Realistic Data Set: Identifying the Origin of Fish from the Growth-Ring Diameter of Scales 35
2.5.3.3: Comparison of Perceptron with Linear Discriminant Function Analysis in Statistics 38

2.5.3.4: Multi-Output Perceptron for Multicategory Classification 40
2.5.3.5: Higher-Dimensional Classification Using Perceptron 45
2.5.3.6: Perceptron Summary 45
2.5.4: Linear Neuron for Linear Classification and Prediction 46
2.5.4.1: Learning with the Delta Rule 47
2.5.4.2: Linear Neuron as a Classifier 51
2.5.4.3: Classification Properties of a Linear Neuron as a Subset of Predictive Capabilities 53
2.5.4.4: Example: Linear Neuron as a Predictor 54
2.5.4.5: A Practical Example of Linear Prediction: Predicting the Heat Influx in a Home 61
2.5.4.6: Comparison of Linear Neuron Model with Linear Regression 62
2.5.4.7: Example: Multiple Input Linear Neuron Model—Improving the Prediction Accuracy of Heat Influx in a Home 63
2.5.4.8: Comparison of a Multiple-Input Linear Neuron with Multiple Linear Regression 63
2.5.4.9: Multiple Linear Neuron Models 64
2.5.4.10: Comparison of a Multiple Linear Neuron Network with Canonical Correlation Analysis 65
2.5.4.11: Linear Neuron and Linear Network Summary 65
2.6: Summary 66
Problems 66
References 67

3 Neural Networks for Nonlinear Pattern Recognition 69
3.1: Overview and Introduction 69
3.1.1: Multilayer Perceptron 71
3.2: Nonlinear Neurons 72
3.2.1: Neuron Activation Functions 73
3.2.1.1: Sigmoid Functions 74
3.2.1.2: Gaussian Functions 76
3.2.2: Example: Population Growth Modeling Using a Nonlinear Neuron 77
3.2.3: Comparison of Nonlinear Neuron with Nonlinear Regression Analysis 80
3.3: One-Input Multilayer Nonlinear Networks 80
3.3.1: Processing with a Single Nonlinear Hidden Neuron 80
3.3.2: Examples: Modeling Cyclical Phenomena with Multiple Nonlinear Neurons 86
3.3.2.1: Example 1: Approximating a Square Wave 86
3.3.2.2: Example 2: Modeling Seasonal Species Migration 94
3.4: Two-Input Multilayer Perceptron Network 98
3.4.1: Processing of Two-Dimensional Inputs by Nonlinear Neurons 98
3.4.2: Network Output 102

3.4.3: Examples: Two-Dimensional Prediction and Classification 103

3.4.3.1: Example 1: Two-Dimensional Nonlinear Function Approximation 103

3.4.3.2: Example 2: Two-Dimensional Nonlinear Classification Model 105

3.5: Multidimensional Data Modeling with Nonlinear Multilayer Perceptron Networks 109

3.6: Summary 110

Problems 110

References 112

4 Learning of Nonlinear Patterns by Neural Networks 113

4.1: Introduction and Overview 113

4.2: Supervised Training of Networks for Nonlinear Pattern Recognition 114

4.3: Gradient Descent and Error Minimization 115

4.4: Backpropagation Learning 116

4.4.1: Example: Backpropagation Training—A Hand Computation 117

4.4.1.1: Error Gradient with Respect to Output Neuron Weights 120

4.4.1.2: The Error Gradient with Respect to the Hidden-Neuron Weights 123

4.4.1.3: Application of Gradient Descent in Backpropagation Learning 127

4.4.1.4: Batch Learning 128

4.4.1.5: Learning Rate and Weight Update 130

4.4.1.6: Example-by-Example (Online) Learning 134

4.4.1.7: Momentum 134

4.4.2: Example: Backpropagation Learning Computer Experiment 138

4.4.3: Single-Input Single-Output Network with Multiple Hidden Neurons 141

4.4.4: Multiple-Input, Multiple-Hidden Neuron, and Single-Output Network 142

4.4.5: Multiple-Input, Multiple-Hidden Neuron, Multiple-Output Network 143

4.4.6: Example: Backpropagation Learning Case Study—Solving a Complex Classification Problem 145

4.5: Delta-Bar-Delta Learning (Adaptive Learning Rate) Method 152

4.5.1: Example: Network Training with Delta-Bar-Delta—A Hand Computation 154

4.5.2: Example: Delta-Bar-Delta with Momentum—A Hand Computation 157

4.5.3: Network Training with Delta-Bar Delta—A Computer Experiment 158

4.5.4: Comparison of Delta-Bar-Delta Method with Backpropagation 159

4.5.5: Example: Network Training with Delta-Bar-Delta—A Case Study 160
4.6: Steepest Descent Method 163
4.6.1: Example: Network Training with Steepest Descent—Hand Computation 163
4.6.2: Example: Network Training with Steepest Descent—A Computer Experiment 164
4.7: Second-Order Methods of Error Minimization and Weight Optimization 166
4.7.1: QuickProp 167
4.7.1.1: Example: Network Training with QuickProp—A Hand Computation 168
4.7.1.2: Example: Network Training with QuickProp—A Computer Experiment 170
4.7.1.3: Comparison of QuickProp with Steepest Descent, Delta-Bar-Delta, and Backpropagation 170
4.7.2: General Concept of Second-Order Methods of Error Minimization 172
4.7.3: Gauss–Newton Method 174
4.7.3.1: Network Training with the Gauss–Newton Method—A Hand Computation 176
4.7.3.2: Example: Network Training with Gauss–Newton Method—A Computer Experiment 178
4.7.4: The Levenberg–Marquardt Method 180
4.7.4.1: Example: Network Training with LM Method—A Hand Computation 182
4.7.4.2: Network Training with the LM Method—A Computer Experiment 183
4.7.5: Comparison of the Efficiency of the First-Order and Second-Order Methods in Minimizing Error 184
4.7.6: Comparison of the Convergence Characteristics of First-Order and Second-Order Learning Methods 185
4.7.6.1: Backpropagation 187
4.7.6.2: Steepest Descent Method 188
4.7.6.3: Gauss–Newton Method 189
4.7.6.4: Levenberg–Marquardt Method 190
4.8: Summary 192
Problems 192
References 193

5 Implementation of Neural Network Models for Extracting Reliable Patterns from Data 195
5.1: Introduction and Overview 195
5.2: Bias–Variance Tradeoff 196
5.3: Improving Generalization of Neural Networks 197
5.3.1: Illustration of Early Stopping 199
5.3.1.1: Effect of Initial Random Weights 203
5.3.1.2: Weight Structure of the Trained Networks 206

5.3.1.3: Effect of Random Sampling 207
5.3.1.4: Effect of Model Complexity: Number of Hidden Neurons 212
5.3.1.5: Summary on Early Stopping 213
5.3.2: Regularization 215
5.4: Reducing Structural Complexity of Networks by Pruning 221
5.4.1: Optimal Brain Damage 222
5.4.1.1: Example of Network Pruning with Optimal Brain Damage 223
5.4.2: Network Pruning Based on Variance of Network Sensitivity 229
5.4.2.1: Illustration of Application of Variance Nullity in Pruning Weights 232
5.4.2.2: Pruning Hidden Neurons Based on Variance Nullity of Sensitivity 235
5.5: Robustness of a Network to Perturbation of Weights 237
5.5.1: Confidence Intervals for Weights 239
5.6: Summary 241
Problems 242
References 243

6 Data Exploration, Dimensionality Reduction, and Feature Extraction 245
6.1: Introduction and Overview 245
6.1.1: Example: Thermal Conductivity of Wood in Relation to Correlated Input Data 247
6.2: Data Visualization 248
6.2.1: Correlation Scatter Plots and Histograms 248
6.2.2: Parallel Visualization 249
6.2.3: Projecting Multidimensional Data onto Two-Dimensional Plane 250
6.3: Correlation and Covariance between Variables 251
6.4: Normalization of Data 253
6.4.1: Standardization 253
6.4.2: Simple Range Scaling 254
6.4.3: Whitening—Normalization of Correlated Multivariate Data 255
6.5: Selecting Relevant Inputs 259
6.5.1: Statistical Tools for Variable Selection 260
6.5.1.1: Partial Correlation 260
6.5.1.2: Multiple Regression and Best-Subsets Regression 261
6.6: Dimensionality Reduction and Feature Extraction 262
6.6.1: Multicollinearity 262
6.6.2: Principal Component Analysis (PCA) 263
6.6.3: Partial Least-Squares Regression 267
6.7: Outlier Detection 268
6.8: Noise 270

6.9: Case Study: Illustrating Input Selection and Dimensionality Reduction for a Practical Problem 270
6.9.1: Data Preprocessing and Preliminary Modeling 271
6.9.2: PCA-Based Neural Network Modeling 275
6.9.3: Effect of Hidden Neurons for Non-PCA- and PCA-Based Approaches 278
6.9.4: Case Study Summary 279
6.10: Summary 280
Problems 281
References 281

7 Assessment of Uncertainty of Neural Network Models Using Bayesian Statistics 283
7.1: Introduction and Overview 283
7.2: Estimating Weight Uncertainty Using Bayesian Statistics 285
7.2.1: Quality Criterion 285
7.2.2: Incorporating Bayesian Statistics to Estimate Weight Uncertainty 288
7.2.2.1: Square Error 289
7.2.3: Intrinsic Uncertainty of Targets for Multivariate Output 292
7.2.4: Probability Density Function of Weights 293
7.2.5: Example Illustrating Generation of Probability Distribution of Weights 295
7.2.5.1: Estimation of Geophysical Parameters from Remote Sensing: A Case Study 295
7.3: Assessing Uncertainty of Neural Network Outputs Using Bayesian Statistics 300
7.3.1: Example Illustrating Uncertainty Assessment of Output Errors 301
7.3.1.1: Total Network Output Errors 301
7.3.1.2: Error Correlation and Covariance Matrices 302
7.3.1.3: Statistical Analysis of Error Covariance 302
7.3.1.4: Decomposition of Total Output Error into Model Error and Intrinsic Noise 304
7.4: Assessing the Sensitivity of Network Outputs to Inputs 311
7.4.1: Approaches to Determine the Influence of Inputs on Outputs in Feedforward Networks 311
7.4.1.1: Methods Based on Magnitude of Weights 311
7.4.1.2: Sensitivity Analysis 312
7.4.2: Example: Comparison of Methods to Assess the Influence of Inputs on Outputs 313
7.4.3: Uncertainty of Sensitivities 314
7.4.4: Example Illustrating Uncertainty Assessment of Network Sensitivity to Inputs 315
7.4.4.1: PCA Decomposition of Inputs and Outputs 315
7.4.4.2: PCA-Based Neural Network Regression 320
7.4.4.3: Neural Network Sensitivities 323
7.4.4.4: Uncertainty of Input Sensitivity 325

7.4.4.5: PCA-Regularized Jacobians 328
7.4.4.6: Case Study Summary 333
7.5: Summary 333
Problems 334
References 335

8 Discovering Unknown Clusters in Data with Self-Organizing Maps 337
8.1: Introduction and Overview 337
8.2: Structure of Unsupervised Networks 338
8.3: Learning in Unsupervised Networks 339
8.4: Implementation of Competitive Learning 340
8.4.1: Winner Selection Based on Neuron Activation 340
8.4.2: Winner Selection Based on Distance to Input Vector 341
8.4.2.1: Other Distance Measures 342
8.4.3: Competitive Learning Example 343
8.4.3.1: Recursive Versus Batch Learning 344
8.4.3.2: Illustration of the Calculations Involved in Winner Selection 344
8.4.3.3: Network Training 346
8.5: Self-Organizing Feature Maps 349
8.5.1: Learning in Self-Organizing Map Networks 349
8.5.1.1: Selection of Neighborhood Geometry 349
8.5.1.2: Training of Self-Organizing Maps 350
8.5.1.3: Neighbor Strength 350
8.5.1.4: Example: Training Self-Organizing Networks with a Neighbor Feature 351
8.5.1.5: Neighbor Matrix and Distance to Neighbors from the Winner 354
8.5.1.6: Shrinking Neighborhood Size with Iterations 357
8.5.1.7: Learning Rate Decay 358
8.5.1.8: Weight Update Incorporating Learning Rate and Neighborhood Decay 359
8.5.1.9: Recursive and Batch Training and Relation to K-Means Clustering 360
8.5.1.10: Two Phases of Self-Organizing Map Training 360
8.5.1.11: Example: Illustrating Self-Organizing Map Learning with a Hand Calculation 361
8.5.1.12: SOM Case Study: Determination of Mastitis Health Status of Dairy Herd from Combined Milk Traits 368
8.5.2: Example of Two-Dimensional Self-Organizing Maps: Clustering Canadian and Alaskan Salmon Based on the Diameter of Growth Rings of the Scales 371
8.5.2.1: Map Structure and Initialization 372
8.5.2.2: Map Training 373
8.5.2.3: U-Matrix 380
8.5.3: Map Initialization 382

8.5.4: Example: Training Two-Dimensional Maps on Multidimensional Data 382
8.5.4.1: Data Visualization 383
8.5.4.2: Map Structure and Training 383
8.5.4.3: U-Matrix 389
8.5.4.4: Point Estimates of Probability Density of Inputs Captured by the Map 390
8.5.4.5: Quantization Error 391
8.5.4.6: Accuracy of Retrieval of Input Data from the Map 393
8.5.5: Forming Clusters on the Map 395
8.5.5.1: Approaches to Clustering 396
8.5.5.2: Example Illustrating Clustering on a Trained Map 397
8.5.5.3: Finding Optimum Clusters on the Map with the Ward Method 401
8.5.5.4: Finding Optimum Clusters by K-Means Clustering 403
8.5.6: Validation of a Trained Map 406
8.5.6.1: *n*-Fold Cross Validation 406
8.6: Evolving Self-Organizing Maps 411
8.6.1: Growing Cell Structure of Map 413
8.6.1.1: Centroid Method for Mapping Input Data onto Positions between Neurons on the Map 416
8.6.2: Dynamic Self-Organizing Maps with Controlled Growth (GSOM) 419
8.6.2.1: Example: Application of Dynamic Self-Organizing Maps 422
8.6.3: Evolving Tree 427
8.7: Summary 431
Problems 432
References 434

9 Neural Networks for Time-Series Forecasting 437
9.1: Introduction and Overview 437
9.2: Linear Forecasting of Time-Series with Statistical and Neural Network Models 440
9.2.1: Example Case Study: Regulating Temperature of a Furnace 442
9.2.1.1: Multistep-Ahead Linear Forecasting 444
9.3: Neural Networks for Nonlinear Time-Series Forecasting 446
9.3.1: Focused Time-Lagged and Dynamically Driven Recurrent Networks 446
9.3.1.1: Focused Time-Lagged Feedforward Networks 448
9.3.1.2: Spatio-Temporal Time-Lagged Networks 450
9.3.2: Example: Spatio-Temporal Time-Lagged Network—Regulating Temperature in a Furnace 452

9.3.2.1: Single-Step Forecasting with Neural NARx Model 454
9.3.2.2: Multistep Forecasting with Neural NARx Model 455
9.3.3: Case Study: River Flow Forecasting 457
9.3.3.1: Linear Model for River Flow Forecasting 460
9.3.3.2: Nonlinear Neural (NARx) Model for River Flow Forecasting 463
9.3.3.3: Input Sensitivity 467
9.4: Hybrid Linear (ARIMA) and Nonlinear Neural Network Models 468
9.4.1: Case Study: Forecasting the Annual Number of Sunspots 470
9.5: Automatic Generation of Network Structure Using Simplest Structure Concept 471
9.5.1: Case Study: Forecasting Air Pollution with Automatic Neural Network Model Generation 473
9.6: Generalized Neuron Network 475
9.6.1: Case Study: Short-Term Load Forecasting with a Generalized Neuron Network 482
9.7: Dynamically Driven Recurrent Networks 485
9.7.1: Recurrent Networks with Hidden Neuron Feedback 485
9.7.1.1: Encapsulating Long-Term Memory 485
9.7.1.2: Structure and Operation of the Elman Network 488
9.7.1.3: Training Recurrent Networks 490
9.7.1.4: Network Training Example: Hand Calculation 495
9.7.1.5: Recurrent Learning Network Application Case Study: Rainfall Runoff Modeling 500
9.7.1.6: Two-Step-Ahead Forecasting with Recurrent Networks 503
9.7.1.7: Real-Time Recurrent Learning Case Study: Two-Step-Ahead Stream Flow Forecasting 505
9.7.2: Recurrent Networks with Output Feedback 508
9.7.2.1: Encapsulating Long-Term Memory in Recurrent Networks with Output Feedback 508
9.7.2.2: Application of a Recurrent Net with Output and Error Feedback and Exogenous Inputs: (NARIMAx) Case Study: Short-Term Temperature Forecasting 510
9.7.2.3: Training of Recurrent Nets with Output Feedback 513
9.7.3: Fully Recurrent Network 515
9.7.3.1: Fully Recurrent Network Practical Application Case Study: Short-Term Electricity Load Forecasting 517
9.8: Bias and Variance in Time-Series Forecasting 519
9.8.1: Decomposition of Total Error into Bias and Variance Components 521
9.8.2: Example Illustrating Bias–Variance Decomposition 522

9.9: Long-Term Forecasting 528
9.9.1: Case Study: Long-Term Forecasting with Multiple Neural Networks (MNNs) 531
9.10: Input Selection for Time-Series Forecasting 533
9.10.1: Input Selection from Nonlinearly Dependent Variables 535
9.10.1.1 Partial Mutual Information Method 535
9.10.1.2 Generalized Regression Neural Network 538
9.10.1.3 Self-Organizing Maps for Input Selection 539
9.10.1.4 Genetic Algorithms for Input Selection 541
9.10.2: Practical Application of Input Selection Methods for Time-Series Forecasting 543
9.10.3: Input Selection Case Study: Selecting Inputs for Forecasting River Salinity 546
9.11: Summary 549
Problems 551
References 552

Appendix 555

Index 561