

1. Record Nr.	UNISA996475764403316
Autore	Wang Liang <1975->
Titolo	OCaml scientific computing : functional programming in data science and artificial intelligence // Liang Wang, Jianxin Zhao, and Richard Mortier
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2022] ©2022
ISBN	3-030-97645-9
Descrizione fisica	1 online resource (372 pages)
Collana	Undergraduate Topics in Computer Science Ser.
Disciplina	005.114
Soggetti	OCaml (Computer program language) Artificial intelligence - Data processing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Intro -- Foreword -- Preface -- Acknowledgements -- Contents -- Acronyms -- Part I Numerical Techniques -- Chapter 1 Introduction -- 1.1 Scientific Computing -- 1.2 Functional Programming -- 1.3 OCaml Numerical Library -- 1.4 System Architecture -- 1.5 How to Run Code -- 1.6 Summary -- References -- Chapter 2 Numerical Algorithms -- 2.1 Interpolation -- 2.1.1 Polynomial Interpolation -- 2.1.2 Rational Interpolation -- 2.2 Integration -- 2.2.1 Basic Algorithms -- 2.2.2 Gaussian Quadrature -- 2.3 Special Functions -- 2.4 Summary -- References -- Chapter 3 Statistics -- 3.1 Random Variables -- 3.1.1 Descriptive Statistics -- 3.2 Distribution -- 3.2.1 Discrete Distribution -- 3.2.2 Continuous Distribution -- 3.2.3 Special Distributions -- 3.3 Multiple Variables -- 3.3.1 Joint and Conditional Probability -- 3.3.2 Covariance and Correlation -- 3.4 Sampling -- 3.4.1 Unbiased Estimator -- 3.4.2 Inferring Population Parameters -- 3.5 Hypothesis Tests -- 3.5.1 Theory -- 3.5.2 Gaussian Distribution in Hypothesis Testing -- 3.5.3 Two-Sample Inferences -- 3.5.4 More Tests -- 3.6 Summary -- References -- Chapter 4 Linear Algebra -- 4.1 Vectors and Matrices -- 4.2 Gaussian Elimination -- 4.2.1 LU Factorisation -- 4.2.2 Inverse and Transpose -- 4.3 Vector Space -- 4.3.1 Rank and Basis -- 4.3.2 Orthogonality -- 4.3.3 Solving $Ax = b$ -- 4.3.4 Matrix Sensitivity -- 4.4 Determinants -- 4.5 Eigenvalues and Eigenvectors -- 4.5.1

Complex Matrices -- 4.5.2 Similarity Transformation and Diagonalisation -- 4.6 Positive Definite Matrices -- 4.6.1 Positive Definiteness -- 4.6.2 Singular Value Decomposition -- 4.7 Sparse Matrices -- 4.8 Summary -- References -- Chapter 5 N-Dimensional Arrays -- 5.1 Narray -- 5.1.1 Types -- 5.1.2 Narray Creation and Properties -- 5.1.3 Map, Fold, and Scan -- 5.1.4 Comparison Functions -- 5.1.5 Iteration Functions.

5.1.6 Manipulation Functions -- 5.1.7 Serialisation -- 5.2 Slicing -- 5.2.1 Slice Definition -- 5.2.2 Conventions and Examples -- 5.2.3 Advanced Usage -- 5.3 Broadcasting -- 5.4 Tensors -- 5.5 Summary -- References -- Chapter 6 Ordinary Differential Equations -- 6.1 Defining an ODE -- 6.1.1 Exact Solutions -- 6.1.2 Linear Systems -- 6.2 Solving ODEs Numerically -- 6.3 ODE Solvers -- 6.3.1 Solving Linear Oscillator System -- 6.3.2 Solver Structure -- 6.3.3 Symplectic Solver -- 6.3.4 Features and Limits -- 6.4 Examples of ODE Solvers -- 6.4.1 Explicit ODE -- 6.4.2 Two-body Problem -- 6.4.3 Lorenz Attractor -- 6.4.4 Damped Oscillation -- 6.5 Stiffness -- 6.5.1 Solving Non-Stiff ODEs -- 6.5.2 Solve Stiff ODEs -- 6.6 Summary -- References -- Chapter 7 Signal Processing -- 7.1 Discrete Fourier Transform -- 7.2 Fast Fourier Transform -- 7.2.1 Example: 1-D Discrete Fourier Transforms -- 7.3 Applications of FFTs -- 7.3.1 Finding the Period of Sunspots -- 7.3.2 Determine the Tone -- 7.3.3 Image Processing -- 7.4 Filtering -- 7.4.1 Example: Smoothing -- 7.4.2 Gaussian Filter -- 7.4.3 Signal Convolution -- 7.4.4 FFT and Image Convolution -- 7.5 Summary -- References -- Part II Advanced Data Analysis Techniques -- Chapter 8 Algorithmic Differentiation -- 8.1 Chain Rule -- 8.2 Differentiation Methods -- 8.2.1 Numerical Differentiation -- 8.2.2 Symbolic Differentiation -- 8.2.3 Algorithmic Differentiation -- 8.3 How Algorithmic Differentiation Works -- 8.3.1 Forward Mode -- 8.3.2 Reverse Mode -- 8.3.3 Forward Mode or Reverse Mode? -- 8.4 A Strawman AD Engine -- 8.4.1 Implementation of Forward Mode -- 8.4.2 Implementation of Reverse Mode -- 8.4.3 A Unified Implementation -- 8.5 Forward and Reverse Propagation API -- 8.5.1 Expressing Computation -- 8.5.2 Example: Forward Mode -- 8.5.3 Example: Reverse Mode -- 8.6 High-Level Functions -- 8.6.1 Derivative and Gradient.

8.6.2 Jacobian -- 8.6.3 Hessian and Laplacian -- 8.6.4 Other APIs -- 8.7 Internals of Algorithmic Differentiation -- 8.7.1 Architecture and Components -- 8.7.2 Extending AD -- 8.7.3 Lazy Evaluation -- 8.8 Summary -- References -- Chapter 9 Optimisation -- 9.1 Objective Functions -- 9.2 Root Finding -- 9.3 Univariate Function Optimisation -- 9.3.1 Use Derivatives -- 9.3.2 Golden Section Search -- 9.4 Multivariate Function Optimisation -- 9.4.1 Nelder-Mead Simplex Method -- 9.4.2 Gradient Descent Methods -- 9.4.3 Conjugate Gradient Method -- 9.4.4 Newton and Quasi-Newton Methods -- 9.5 Global Optimisation and Constrained Optimisation -- 9.6 Summary -- References -- Chapter 10 Regression -- 10.1 Linear Regression -- 10.1.1 Problem: Where to open a new McDonald's restaurant? -- 10.1.2 Cost Function -- 10.1.3 Solving Problem with Gradient Descent -- 10.2 Multiple Regression -- 10.2.1 Feature Normalisation -- 10.2.2 Analytical Solution -- 10.3 Non-Linear Regressions -- 10.4 Regularisation -- 10.4.1 Ols, Ridge, Lasso, and Elastic\_net -- 10.5 Logistic Regression -- 10.5.1 Sigmoid Function -- 10.5.2 Cost Function -- 10.5.3 Example -- 10.5.4 Multi-class Classification -- 10.6 Support Vector Machines, SVMs -- 10.6.1 Kernel and Non-Linear Boundary -- 10.6.2 Example -- 10.7 Model Error and Selection -- 10.7.1 Error Metrics -- 10.7.2 Model Selection -- 10.8 Summary -- References -- Chapter 11 Neural Network -- 11.1 The Perceptron --

11.2 Yet Another Regression -- 11.2.1 Model Representation -- 11.2.2 Forward Propagation -- 11.2.3 Back Propagation -- 11.2.4 Feedforward Network -- 11.2.5 Layers -- 11.2.6 Activation Functions -- 11.2.7 Initialisation -- 11.2.8 Training -- 11.2.9 Test -- 11.3 Neural Network Module -- 11.3.1 Neurons -- 11.3.1.1 Neural Graph -- 11.3.2 Training Parameters -- 11.4 Convolutional Neural Network -- 11.5 Recurrent Neural Network. 11.6 Generative Adversarial Network -- 11.7 Summary -- References -- Chapter 12 Vector Space Modelling -- 12.1 Introduction -- 12.2 Text Corpus Analysis -- 12.2.1 Building a Text Corpus -- 12.2.2 Using Corpus Module -- 12.3 Vector Space Models -- 12.3.1 Word Embedding and BERT -- 12.3.2 Bag of Words (BOW) -- 12.4 Term Frequency-Inverse Document Frequency (TF-IDF) -- 12.5 Latent Dirichlet Allocation (LDA) -- 12.5.1 Models -- 12.5.2 Dirichlet Distribution -- 12.5.3 Gibbs Sampling -- 12.5.4 Topic Modelling Example -- 12.6 Latent Semantic Analysis -- 12.7 Search Relevant Documents -- 12.7.1 Euclidean and Cosine Similarity -- 12.7.2 Linear Searching -- 12.8 Summary -- References -- Part III Use Cases -- Chapter 13 Case Study: Image Recognition -- 13.1 Types of Networks -- 13.1.1 LeNet -- 13.1.2 AlexNet -- 13.1.3 VGG -- 13.1.4 ResNet -- 13.1.5 SqueezeNet -- 13.1.6 Capsule Network -- 13.2 Building the InceptionV3 Network -- 13.2.1 InceptionV1 and InceptionV2 -- 13.2.2 Factorisation -- 13.2.3 Grid Size Reduction -- 13.2.4 InceptionV3 Architecture -- 13.3 Preparing Weights -- 13.4 Processing Image -- 13.5 Running Inference -- 13.6 Applications -- 13.7 Summary -- References -- Chapter 14 Case Study: Instance Segmentation -- 14.1 Introduction -- 14.2 Object Detection Architectures -- 14.2.1 R-CNN -- 14.2.2 Fast R-CNN -- 14.2.3 Faster R-CNN -- 14.2.4 Mask R-CNN -- 14.3 Mask R-CNN Network -- 14.3.1 Building Mask R-CNN -- 14.3.2 Feature Extractor -- 14.3.3 Proposal Generation -- 14.3.4 Classification -- 14.4 Run the Code -- 14.5 Summary -- References -- Chapter 15 Case Study: Neural Style Transfer -- 15.1 Content and Style -- 15.1.1 Content Reconstruction -- 15.1.2 Style Recreation -- 15.1.3 Combining Content and Style -- 15.1.4 Running NST -- 15.2 Extending NST -- 15.3 Fast Style Transfer -- 15.3.1 Building FST Network -- 15.3.2 Running FST -- 15.4 Summary. References -- Chapter 16 Case Study: Recommender System -- 16.1 Introduction -- 16.2 Key Components And Pipeline -- 16.3 Reducing Dimensionality -- 16.4 Random Projection -- 16.5 Vector Storage Optimisation -- 16.6 Combining Multiple Trees -- 16.7 Critical Boundary Elimination -- 16.8 Search Operations Parallelisation -- 16.9 Code Implementation -- 16.10 Summary -- References -- Appendix A Conventions of Owl -- A.1 Pure vs Impure -- A.2 Narray vs Scalar -- A.3 Infix Operators -- A.4 Module Structures -- A.5 Operator Extension -- Appendix B Visualisation -- B.1 Plotting in Owl -- B.1.1 Create Plots -- B.1.2 Specification -- B.1.3 Subplots -- B.1.4 Multiple Lines -- B.1.5 Legend -- B.1.6 Drawing Patterns -- B.2 Plot Types -- Index.

---