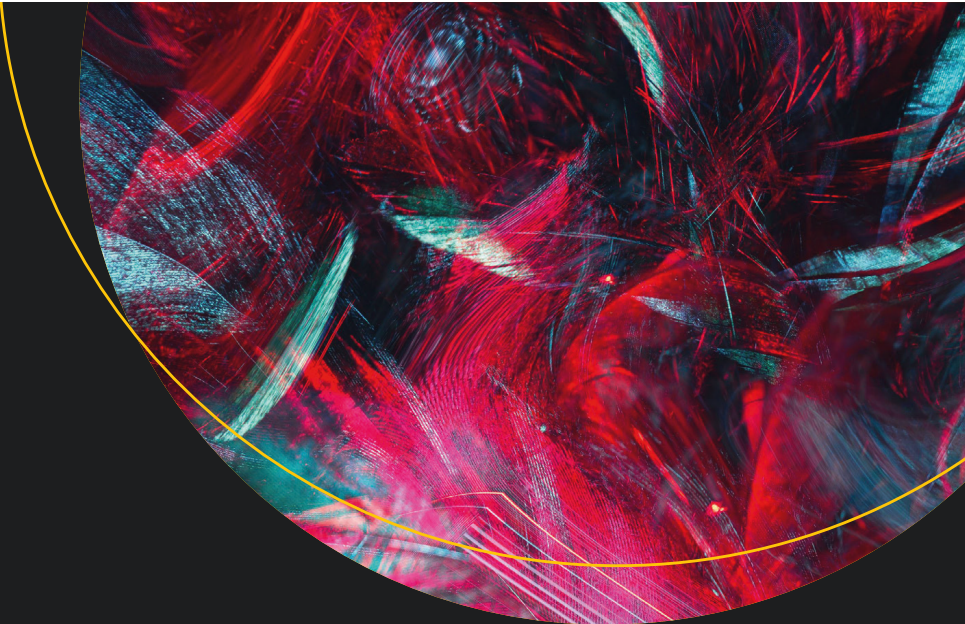




Practical Artificial Intelligence in Julia

Implementing Neural Networks, Genetic Algorithms, and Neuroevolution in Julia

Alexandre Bergel

Apress®

Practical Artificial Intelligence in Julia

**Implementing Neural Networks,
Genetic Algorithms, and
Neuroevolution in Julia**

Alexandre Bergel

Apress®

Practical Artificial Intelligence in Julia: Implementing Neural Networks, Genetic Algorithms, and Neuroevolution in Julia

Alexandre Bergel
German Aerospace Center (DLR),
Institute of Software Technology, 82234,
Wessling, Bavaria, Germany

ISBN-13 (pbk): 979-8-8688-2594-1
<https://doi.org/10.1007/979-8-8688-2595-8>

ISBN-13 (electronic): 979-8-8688-2595-8

Copyright © 2026 by Alexandre Bergel

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Melissa Duffy
Editorial Project Manager: Gryffin Winkler

Cover designed by eStudioCalamar

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a Delaware LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub. For more detailed information, please visit <https://www.apress.com/gp/services/source-code>.

If disposing of this product, please recycle the paper

Table of Contents

About the Author	ix
About the Technical Reviewer	xi
Introduction	xiii
Part I: Classifying Data with Neural Networks	1
Chapter 1: The Perceptron Model	3
Perceptron As a Kind of Neuron	3
Implementing the Perceptron	5
Testing Our Code.....	7
Formulating Logical Expressions	8
Error Handling	10
Combining Perceptrons.....	11
Training a Perceptron.....	13
Drawing Graphs	18
Number of Epochs and Precision	20
Predicting 2D Points.....	22
Measuring the Precision	27
Historical Perspective	28
Exercises.....	28
What Have We Seen?	29
Chapter 2: Sigmoid Neuron.....	31
Limit of a Single Perceptron.....	31
Activation Function	32
The Sigmoid Neuron	33
Extending the Neuron with the Activation Functions	34

TABLE OF CONTENTS

Testing the Sigmoid Neuron	37
Slower to Learn	40
What Have We Seen in This Chapter?	41
Chapter 3: Neural Networks	43
General Architecture	43
Neural Layer	44
Modeling a Neural Network	48
Backpropagation	50
Step 1: Forward Feeding	51
Step 2: Error Backward Propagation	52
Step 3: Updating Neurons' Parameters	53
What Have We Seen in This Chapter?	56
Chapter 4: Theory on Learning	57
Loss Function	57
Gradient Descent	60
Parameter Update	63
What Have We Seen in This Chapter?	66
Further Reading	67
Chapter 5: Data Classification	69
Train a Network	69
Neural Network As a Hashmap	73
Visualizing the Error and the Prediction Accuracy	74
Contradictory Data	77
Classifying Data and One Hot Encoding	78
Iris Dataset	79
Training a Network with the Iris Dataset	81
Effect of the Learning Curve	83
Test and Validation	85

Normalization.....	89
Integrating the Normalization into a NNetwork	94
What Have We Seen in This Chapter?	97
Part II: Solving Problems and Defining Creature Behavior with Genetic Algorithms	99
Chapter 6: Genetic Algorithms.....	101
Algorithms Inspired from Natural Evolution	101
Example of a Genetic Algorithm.....	102
Relevant Vocabulary.....	103
Modeling Individual.....	104
Crossover Genetic Operation.....	110
Mutation Genetic Operation	112
Parent Selection.....	114
Monitoring the Evolution and Termination	117
Terminating the Algorithm.....	118
Running the Genetic Algorithm	121
Our Algorithm in Action	123
What Have We Seen in This Chapter?	125
Chapter 7: Genetic Algorithm in Action	127
Visualizing the Evolution	127
Mutation Rate vs. Population Size	131
Fundamental Theorem of Arithmetic.....	133
Knapsack Problem	135
The Unbounded Knapsack Problem Variant.....	136
The 0-1 Knapsack Problem Variant	138
Coding and Encoding.....	140
Meeting Room Scheduling Problem.....	140
Mini Sudoku	142
Exploration and Exploitation	145

TABLE OF CONTENTS

Adaptive Mutation Strategies	147
What Have We Seen in This Chapter?	147
Acknowledgment	148
Chapter 8: The Traveling Salesman Problem	149
Illustration of the Problem.....	149
Relevance of the Traveling Salesman Problem	150
Naive Approach	151
The Roassal Graphic System	154
Naive Approach on a Larger Graph	156
Adequate Genetic Operations.....	159
Swap Mutation Operation	160
Ordered Crossover Operation	161
Revisiting Our Large Example	163
What Have We Seen in This Chapter?	165
Chapter 9: Building Walking Creatures.....	167
Modeling Joint Points	168
Modeling Platforms.....	171
Modeling Muscle.....	174
Generating Muscles	176
Modeling Creatures.....	178
Creating Creatures	180
Serialization and Materialization of a Creature	182
Accessors and Utility Functions	183
Defining the World.....	184
Cold Run.....	187
What Have We Seen in This Chapter?	188

Chapter 10: Evolving Walking Creatures	189
Straightforward Application of Genetic Algorithm.....	189
Take Inspiration from Nature to Address the Competing Conventions Problem	193
Constrained Crossover Operation	194
Moving Forward	195
Serializing the Muscle Attributes	197
Climbing Stairs.....	198
Alternative Fitness Function	200
What Have We Seen in This Chapter?	200
Part III: Playing A Platform Game With Neuroevolution.....	201
Chapter 11: Neuroevolution	203
Supervised Learning, Unsupervised Learning, Reinforcement Learning	203
Neuroevolution.....	204
Two Techniques for Neuroevolution	205
The NeuroGenetic Approach	206
Neural Network Serialization and Materialization.....	207
Evolving Weights and Biases	209
Converting Binary Values to Decimal	211
Iris Dataset.....	214
Further Reading About NeuroGenetic.....	217
What Have We Seen in This Chapter?	217
Chapter 12: Neuroevolution with NEAT	219
Vocabulary	220
Node.....	221
Different Kind of Nodes	222
Connections	225
Individual.....	226

TABLE OF CONTENTS

Networking Building	229
Species	233
Speciation	234
Crossover Operation.....	236
Mutating by Adding a Connection.....	238
Mutating by Adding a Node	240
Mutating by Changing a Connection Weight.....	242
Logging the Evolution	242
NEAT.....	243
Visualizing the Evolution	252
The XOR Example.....	253
The Iris Example	256
Loading and Saving.....	258
What Have We Seen in This Chapter?	260
Chapter 13: The Platform Video Game.....	261
Dependencies	262
Game Assets	263
Monster Definition.....	265
Hero Definition	269
Modeling the World	272
Abstracting the Game Representation	279
Playing the Platform Game	282
Plugging an Agent.....	283
What Have We Seen in This Chapter?	284
Chapter 14: Neuroevolution on the Platform Game	285
Building an Artificial Player	285
Memoryless Game	288
What Have We Seen in This Chapter?	288
Index.....	289

About the Author



Dr. Alexandre Bergel has been a senior researcher at the German Aerospace Center (DLR) since 2026. Previously, he was a computer scientist at RelationalAI (Switzerland/USA) and an associate professor at the University of Chile. Dr. Bergel and his collaborators conduct research in software engineering and artificial intelligence for critical software systems.

He has authored over 180 articles published in international, peer-reviewed scientific forums, including the most competitive conferences and journals in software engineering. He has served on over 185 program committees for international events. Several of his research prototypes have been developed into products adopted by major companies in the semiconductor industry and for software certification.

He is the author of four other books: *Agile Visualization with Pharo*, *Agile Artificial Intelligence in Pharo*, *Agile Visualization*, and *Deep into Pharo*.

About the Technical Reviewer



Stefania Loredana Nita, PhD, is a lecturer at the “Ferdinand I” Military Technical Academy of Bucharest and a software developer and researcher at the Institute for Computers. Her PhD thesis was on advanced cryptographic schemes using searchable encryption and homomorphic encryption. At the Military Technical Academy, she teaches courses in formal languages and translators and database application development. She has worked for more than two years as an assistant lecturer at the University of Bucharest, where she taught courses on subjects such as advanced programming techniques, simulation methods, and operating systems. Her

research activity is in the cryptography field, with a focus on searchable encryption and homomorphic encryption. She is also interested in blockchain, quantum cryptography, machine learning, and artificial intelligence. At the Institute for Computers, she is working on research and development projects that involve cloud computing security, the Internet of Things, and big data. She has authored and co-authored more than 40 papers at conferences and in journals and has co-authored ten books. She holds an MSc degree in software engineering and two BSc degrees, one in computer science and one in mathematics.

Introduction

Artificial intelligence (AI) radically changes how we use computers to solve problems. By exploiting previous experience, which may be expressed in terms of examples, the machine can identify patterns in a given situation and identify the same patterns in a slightly different situation. This is essentially the way AI is used nowadays. The field of AI is moving fast, and unfortunately, it is often difficult to make its way through.

Practical Artificial Intelligence in Julia aims to provide a practical foundation for a set of expressive artificial intelligence algorithms using the Julia programming language. The book makes two large contributions to existing related books. The first contribution details some expressive techniques, namely, neural networks, genetic algorithms, and neuroevolution. No other books comprehensively relate to these topics by providing a full Julia implementation. The book provides material in an incremental fashion, beginning with a little perceptron and ending with a full implementation of two algorithms for neuroevolution. The second contribution is to provide a set of compelling, non-trivial, and self-contained applications of these techniques, including AI for a platform game and a computational model of evolving walking creatures.

The book is meant to be accessible to a large audience and delivers executable source code. The implementation provided in this book is self-contained and relies only on common low-level libraries.

Overall, this book details and illustrates some easy-to-use recipes for solving concrete and practical problems. It provides a complete and reusable set of tools in the Julia programming language. *Practical Artificial Intelligence in Julia* is not a book about how to use an existing API provided by external libraries. Instead, this book guides you to build your own API for artificial intelligence. Building your API has incredible benefits, such as high tunability and malleability, which are essential to solving very specific problems.

Book Overview

Practical Artificial Intelligence in Julia is divided into three parts, each targeting a specific topic within the field of artificial intelligence, namely, neural networks, genetic algorithms, and neuroevolution.

INTRODUCTION

The first part of the book is about *neural networks*. A neural network is a computational metaphor simulating the interaction occurring between biological neurons. The chapter begins with the implementation of a single neuron and shows its limits in terms of what it can achieve. Neural networks are then presented to solve a number of problems. Various examples involving relatively simple data classification tasks are presented.

The second part of the book details *Genetic Algorithms* (GAs). From a technical perspective, GA is about optimizing a fitness function by applying biologically inspired operators. As a result, GA solves problems without telling the structure and shape of the solution in advance. GA simulates the way biological species evolve over time by combining genetic data. For two candidate solutions, as soon as there is a way to say which one is closer to the solution, GA may be considered to solve the problem. Numerous examples are provided in this second part of the book, including an implementation of walking creatures, which is an artificial life simulation. Such creatures are artificial organisms able to evolve in order to move themselves through obstacles or climb stairs.

The third part of the book covers the field of *Neuroevolution*, which is a combination of genetic algorithms and neural networks. The evolution of neural networks is called Neuroevolution. Instead of *training* a neural network, as in classical deep learning (Part 1 of the book), neuroevolution begins with extremely simple networks and incrementally adds complexity to these networks. Evolution makes evolved networks able to solve particularly complex tasks. This third part of the book uses a platform video game and builds an artificial player using neuroevolution.

The Julia Programming Language

Julia is a programming language that is enjoying rising popularity. Julia is simple to use, supports dynamic and static typing, and provides many tools to support interactive programming.

The Julia programming language operates on most platforms and is easy to install. The webpage <https://julialang.org/> provides detailed instructions on how to install Julia on your favorite operating system. The book's content is known to work with all recent Julia versions.

Accompanying Source Code

Practical Artificial Intelligence in Julia is a book about programming. It details a number of non-trivial algorithms and provides a sizable amount of source code. Most of the code given in the book is self-contained. This means that no external libraries are used besides the one of Julia. The only exception is Roassal, a small graphic API that we will use to build animations.

Readers may opt for transcribing the code into Julia or using our dedicated Git repository: <https://github.com/bergel/PracticalAIJulia-Code>.

Julia comes with a full-featured interactive command-line REPL (Read-Eval-Print loop), which this book intensively uses. The Julia REPL is accessible by simply running Julia from the command line.

This repository follows a natural structure: the code provided by each chapter is contained in a numbered single file. *All* the source code and snippets given in the book may be found in these files. The code provided in a chapter N assumes that code from the chapter $N - 1$ is executed. For example, before executing the code of Chapter 2, you need to execute the code of Chapter 1. The code given in Chapter 2 can therefore be run in a Julia REPL using:

```
include("01-Perceptron.jl")
include("02-Neuron.jl")
```

The whole source code can be simply loaded in a Julia REPL using:

```
include("all.jl")
```

Note that it frequently happens that a function given in a chapter is overridden in a later chapter. As a consequence, code given in a particular chapter may not work after executing `all.jl`.

Some function definitions can be long, and in that case, individual portions of source code may be discussed. A script that begins with ellipses (i.e., `...`) means that you need to append the script to the last one seen before.

The code could be easily used with the Pluto programming environment. We did not pursue this path to keep the book scoped to its main focus.

Who Should Read This Book?

This book is designed for a wide audience of programmers. As such, prior knowledge of neural networks, genetic algorithms, or neuroevolution is not required, nor is a strong mathematical background. We made sure that there is no such prerequisite for most of the chapters. Some chapters require a mild mathematical background. However, these chapters are self-contained, and skipping them will not negatively affect overall understanding.

The book exposes some sophisticated AI techniques through Julia's lens. Readers will acquire the theoretical and practical tools to be used in Julia. Note that people willing to learn Julia through AI are encouraged to complement it with some additional source of information. *Practical Artificial Intelligence in Julia* is made for programmers familiar with Julia or experienced with programming in general. The reader is expected to invest in programming to follow the book.

The book gives a relatively superficial introduction to neural networks, just enough (i) to have an accurate feeling of what neural networks are about and (ii) to provide a solid base to model neuroevolution. The first part conveys the essence of neural networks and presents some simple applications, including data classifications. However, the book does not mention advanced neural network architecture such as convolution, transformers, or language models. The readers are kindly invited to refer to dedicated books to deepen these topics. The neural network material is provided as a foundation to build the neuroevolution algorithm.

The book goes into detail about genetic algorithms and neuroevolution. Genetic algorithms are highly expressive and can solve complex search problems that neural networks cannot consider. Genetic algorithms can search for a needle in a haystack. An example of this ability to find a hard-to-find solution is to find the topology and parameters of a neural network without using gradient-based learning techniques. This is what the third part of the book is all about.

After reading this book, the readers should have a solid expertise in genetic algorithms and neuroevolution and will be able to apply them in non-trivial situations.

Acknowledgments

Practical Artificial Intelligence in Julia is the result of a long effort initiated in our previous book, *Agile Artificial Intelligence in Pharo*. Our primary acknowledgments go to the Pharo community for their feedback.

We also thank the Apress team for trusting in the project and the assigned technical reviewer for reviewing the book. They did an amazing job!