

Hands-On Genetic Algorithms with Python

Applying genetic algorithms to solve real-world deep learning and artificial intelligence problems



Packt>

www.packt.com

Eyal Wirsansky

Hands-On Genetic Algorithms with Python

Applying genetic algorithms to solve real-world deep learning and artificial intelligence problems

Eyal Wirsansky



BIRMINGHAM - MUMBAI

Hands-On Genetic Algorithms with Python

Copyright © 2020 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing or its dealers and distributors, will be held liable for any damages caused or alleged to have been caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

Commissioning Editor: Sunith Shetty
Acquisition Editor: Porous Godhaa
Content Development Editor: Pratik Andrade
Senior Editor: Ayaan Hoda
Technical Editor: Mohd Riyan Khan
Copy Editor: Safis Editing
Project Coordinator: Anish Daniel
Proofreader: Safis Editing
Indexer: Priyanka Dhadke
Production Designer: Deepika Naik

First published: January 2020

Production reference: 1300120

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham
B3 2PB, UK.

ISBN 978-1-83855-774-4

www.packt.com

*To my wife, Jackie, for her love, patience, and support. To my children, Danielle and Liam,
whose creativity and artistic talents inspired me in writing this book.*

– Eyal Wirsansky



Packt.com

Subscribe to our online digital library for full access to over 7,000 books and videos, as well as industry leading tools to help you plan your personal development and advance your career. For more information, please visit our website.

Why subscribe?

- Spend less time learning and more time coding with practical eBooks and Videos from over 4,000 industry professionals
- Improve your learning with Skill Plans built especially for you
- Get a free eBook or video every month
- Fully searchable for easy access to vital information
- Copy and paste, print, and bookmark content

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.packt.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at customercare@packtpub.com for more details.

At www.packt.com, you can also read a collection of free technical articles, sign up for a range of free newsletters, and receive exclusive discounts and offers on Packt books and eBooks.

Contributors

About the author

Eyal Wirsansky is a senior software engineer, a technology community leader, and an artificial intelligence enthusiast and researcher. Eyal started his software engineering career as a pioneer in the field of voice over IP, and he now has over 20 years' experience of creating a variety of high-performing enterprise solutions. While in graduate school, he focused his research on genetic algorithms and neural networks. One outcome of his research is a novel supervised machine learning algorithm that combines the two.

Eyal leads the Jacksonville (FL) Java user group, hosts the Artificial Intelligence for Enterprise virtual user group, and writes the developer-oriented artificial intelligence blog, [ai4java](http://ai4java.com).

I would like to thank my family and close friends for their patience, support, and encouragement throughout the lengthy process of writing this book. Special thanks go to the Jacksonville Python Users Group (PyJax) for their feedback and support.

About the reviewer

Lisa Bang did her BS in marine biology at UC Santa Cruz, and an MS in bioinformatics at Soongsil University in Seoul under the tutelage of Dr. Kwang-Hwi Cho. Her masters' thesis was on a method for making QSARs reproducible using Jupyter Notebook, and contained a genetic algorithm component to reduce search space. This is now being developed into DEAP-VS to be compatible with Python 3. She also worked at Geisinger Health System as part of the Biomedical and Translational Informatics Institute, using next-generation sequencing and electronic health record data to analyze outcomes in cancer and other diseases. She now works at Ultragenyx Pharmaceutical, focusing on preclinical research using bioinformatics and chemoinformatics on rare genetic diseases.

Thank you to my family, my teachers, and my mentors.

Packt is searching for authors like you

If you're interested in becoming an author for Packt, please visit authors.packtpub.com and apply today. We have worked with thousands of developers and tech professionals, just like you, to help them share their insight with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Table of Contents

Preface	1
Section 1: The Basics of Genetic Algorithms	
Chapter 1: An Introduction to Genetic Algorithms	8
What are genetic algorithms?	9
Darwinian evolution	9
The genetic algorithms analogy	10
Genotype	10
Population	11
Fitness function	11
Selection	11
Crossover	12
Mutation	12
The theory behind genetic algorithms	13
The schema theorem	14
Differences from traditional algorithms	15
Population-based	16
Genetic representation	16
Fitness function	16
Probabilistic behavior	17
Advantages of genetic algorithms	17
Global optimization	18
Handling complex problems	19
Handling a lack of mathematical representation	19
Resilience to noise	19
Parallelism	20
Continuous learning	20
Limitations of genetic algorithms	20
Special definitions	21
Hyperparameter tuning	21
Computationally-intensive	21
Premature convergence	21
No guaranteed solution	22
Use cases of genetic algorithms	22
Summary	23
Further reading	23
Chapter 2: Understanding the Key Components of Genetic Algorithms	24
Basic flow of a genetic algorithm	25
Creating the initial population	26

Calculating the fitness	26
Applying selection, crossover, and mutation	26
Checking the stopping conditions	27
Selection methods	28
Roulette wheel selection	28
Stochastic universal sampling	29
Rank-based selection	30
Fitness scaling	32
Tournament selection	34
Crossover methods	34
Single-point crossover	35
Two-point and k-point crossover	35
Uniform crossover	36
Crossover for ordered lists	37
Ordered crossover	37
Mutation methods	39
Flip bit mutation	40
Swap mutation	40
Inversion mutation	40
Scramble mutation	41
Real-coded genetic algorithms	41
Blend crossover	42
Simulated binary crossover	44
Real mutation	46
Understanding elitism	47
Niching and sharing	47
Serial niching versus parallel niching	49
The art of solving problems using genetic algorithms	50
Summary	51
Further reading	52
Section 2: Solving Problems with Genetic Algorithms	
Chapter 3: Using the DEAP Framework	54
Technical requirements	55
Introduction to DEAP	55
Using the creator module	56
Creating the Fitness class	57
Defining the fitness strategy	57
Storing the fitness values	58
Creating the Individual class	59
Using the Toolbox class	59
Creating genetic operators	60
Creating the population	61

Calculating the fitness	62
The OneMax problem	63
Solving the OneMax problem with DEAP	63
Choosing the chromosome	63
Calculating the fitness	64
Choosing the genetic operators	64
Setting the stopping condition	64
Implementing with DEAP	65
Setting up	65
Evolving the solution	69
Running the program	73
Using built-in algorithms	74
The Statistics object	74
The algorithm	75
The logbook	75
Running the program	76
Adding the hall of fame	77
Experimenting with the algorithm's settings	79
Population size and number of generations	80
Crossover operator	82
Mutation operator	84
Selection operator	86
Tournament size and relation to mutation probability	87
Roulette wheel selection	91
Summary	93
Further reading	93
Chapter 4: Combinatorial Optimization	94
Technical requirements	95
Search problems and combinatorial optimization	95
Solving the knapsack problem	96
The Rosetta Code knapsack 0-1 problem	97
Solution representation	98
Python problem representation	98
Genetic algorithms solution	100
Solving the TSP	102
TSPLIB benchmark files	104
Solution representation	105
Python problem representation	105
Genetic algorithms solution	107
Improving the results with enhanced exploration and elitism	111
Solving the VRP	115
Solution representation	117
Python problem representation	118
Genetic algorithms solution	120

Summary	125
Further reading	125
Chapter 5: Constraint Satisfaction	126
Technical requirements	127
Constraint satisfaction in search problems	127
Solving the N-Queens problem	128
Solution representation	129
Python problem representation	131
Genetic algorithms solution	132
Solving the nurse scheduling problem	137
Solution representation	137
Hard constraints versus soft constraints	138
Python problem representation	140
Genetic algorithms solution	142
Solving the graph coloring problem	146
Solution representation	148
Using hard and soft constraints for the graph coloring problem	149
Python problem representation	149
Genetic algorithms solution	151
Summary	156
Further reading	157
Chapter 6: Optimizing Continuous Functions	158
Technical requirements	158
Chromosomes and genetic operators for real numbers	159
Using DEAP with continuous functions	161
Optimizing the Eggholder function	162
Optimizing the Eggholder function with genetic algorithms	163
Improving the speed with an increased mutation rate	167
Optimizing Himmelblau's function	169
Optimizing Himmelblau's function with genetic algorithms	170
Using niching and sharing to find multiple solutions	174
Simionescu's function and constrained optimization	178
Constrained optimization with genetic algorithms	180
Optimizing Simionescu's function using genetic algorithms	181
Using constraints to find multiple solutions	182
Summary	184
Further reading	184
Section 3: Artificial Intelligence Applications of Genetic Algorithms	
<hr/>	
Chapter 7: Enhancing Machine Learning Models Using Feature Selection	186

Technical requirements	187
Supervised machine learning	187
Classification	189
Regression	190
Supervised learning algorithms	191
Feature selection in supervised learning	192
Selecting the features for the Friedman-1 regression problem	193
Solution representation	194
Python problem representation	194
Genetic algorithms solution	197
Selecting the features for the classification Zoo dataset	199
Python problem representation	201
Genetic algorithms solution	203
Summary	206
Further reading	206
Chapter 8: Hyperparameter Tuning of Machine Learning Models	207
Technical requirements	208
Hyperparameters in machine learning	208
Hyperparameter tuning	210
The Wine dataset	210
The adaptive boosting classifier	211
Tuning the hyperparameters using a genetic grid search	212
Testing the classifier's default performance	214
Running the conventional grid search	215
Running the genetic algorithm-driven grid search	215
Tuning the hyperparameters using a direct genetic approach	216
Hyperparameter representation	217
Evaluating the classifier accuracy	218
Tuning the hyperparameters using genetic algorithms	219
Summary	222
Further reading	222
Chapter 9: Architecture Optimization of Deep Learning Networks	223
Technical requirements	224
Artificial neural networks and deep learning	224
Multilayer Perceptron	226
Deep learning and convolutional neural networks	227
Optimizing the architecture of a deep learning classifier	228
The Iris flower dataset	228
Representing the hidden layer configuration	229
Evaluating the classifier's accuracy	230
Optimizing the MLP architecture using genetic algorithms	232
Combining architecture optimization with hyperparameter tuning	235
Solution representation	236

Evaluating the classifier's accuracy	236
Optimizing the MLP's combined configuration using genetic algorithms	237
Summary	239
Further reading	239
Chapter 10: Reinforcement Learning with Genetic Algorithms	240
Technical requirements	241
Reinforcement learning	241
Genetic algorithms and reinforcement learning	243
OpenAI Gym	243
The env interface	244
Solving the MountainCar environment	245
Solution representation	247
Evaluating the solution	248
Python problem representation	248
Genetic algorithms solution	249
Solving the CartPole environment	252
Controlling the CartPole with a neural network	254
Solution representation and evaluation	255
Python problem representation	256
Genetic algorithms solution	257
Summary	261
Further reading	261
Section 4: Related Technologies	
<hr/>	
Chapter 11: Genetic Image Reconstruction	263
Technical requirements	264
Reconstructing images with polygons	264
Image processing in Python	265
Python image processing libraries	265
The Pillow library	265
The scikit-image library	265
The opencv-python library	266
Drawing images with polygons	267
Measuring the difference between images	268
Pixel-based Mean Squared Error	269
Structural Similarity (SSIM)	269
Using genetic algorithms to reconstruct images	270
Solution representation and evaluation	270
Python problem representation	271
Genetic algorithm implementation	272
Adding a callback to the genetic run	275
Image reconstruction results	277
Using pixel-based Mean Squared Error	278
Using the SSIM index	280

Other experiments	283
Summary	285
Further reading	285
Chapter 12: Other Evolutionary and Bio-Inspired Computation	
Techniques	286
Technical requirements	287
Evolutionary computation and bio-inspired computing	287
Genetic programming	288
Genetic programming example – even parity check	290
Genetic programming implementation	291
Simplifying the solution	298
Particle swarm optimization	300
PSO example – function optimization	301
Particle swarm optimization implementation	302
Other related techniques	307
Evolution strategies	307
Differential evolution	307
Ant colony optimization	308
Artificial immune systems	308
Artificial life	309
Summary	309
Further reading	310
Other Books You May Enjoy	311
Index	314

Preface

Drawing inspiration from Charles Darwin's theory of natural evolution, *genetic algorithms* are among the most fascinating techniques for solving search, optimization, and learning problems. They can often prove successful where traditional algorithms fail to provide adequate results within a reasonable timeframe.

This book will take you on a journey to mastering this extremely powerful, yet simple, approach, and applying it to a wide variety of tasks, culminating in AI applications.

Using this book, you will gain an understanding of genetic algorithms, how they work, and when to use them. In addition, the book will provide you with hands-on experience of applying genetic algorithms to various domains using the popular Python programming language.

Who this book is for

This book was written to help software developers, data scientists, and AI enthusiasts interested in harnessing genetic algorithms to carry out tasks involving learning, searching, and optimization in their applications, as well as enhancing the performance and accuracy of their existing intelligent applications.

This book is also intended for anyone who is tasked with real-life, hard-to-solve problems where traditional algorithms are not useful, or fail to provide adequate results within a practical amount of time. The book demonstrates how genetic algorithms can be used as a powerful, yet simple, approach to solving a variety of complex problems.

What this book covers

Chapter 1, *An Introduction to Genetic Algorithms*, introduces genetic algorithms, their underlying theory, and their basic principles of operation. You will then explore the differences between genetic algorithms and traditional methods, and learn about the best use cases for genetic algorithms.

Chapter 2, *Understanding the Key Components of Genetic Algorithms*, dives deeper into the key components and the implementation details of genetic algorithms. After outlining the basic genetic flow, you will learn about their different components and the various implementations for each component.

Chapter 3, *Using the DEAP Framework*, introduces DEAP—a powerful and flexible evolutionary computation framework capable of solving real-life problems using genetic algorithms. You will discover how to use this framework by writing a Python program that solves the OneMax problem—the 'Hello World' of genetic algorithms.

Chapter 4, *Combinatorial Optimization*, covers combinatorial optimization problems, such as the knapsack problem, the traveling salesman problem, and the vehicle routing problem, and how to write Python programs that solve them using genetic algorithms and the DEAP framework.

Chapter 5, *Constraint Satisfaction*, introduces constraint satisfaction problems, such as the N-Queen problem, the nurse scheduling problem, and the graph coloring problem, and explains how to write Python programs that solve them using genetic algorithms and the DEAP framework.

Chapter 6, *Optimizing Continuous Functions*, covers continuous optimization problems, and how they can be solved by means of genetic algorithms. The examples you will use include the optimization of the Eggholder function, Himmelblau's function, and Simionescu's function. Along the way, you will explore the concepts of niching, sharing, and constraint handling.

Chapter 7, *Enhancing Machine Learning Models Using Feature Selection*, talks about supervised machine learning models, and explains how genetic algorithms can be used to improve the performance of these models by selecting the best subset of features from the input data provided.

Chapter 8, *Hyperparameter Tuning of Machine Learning Models*, explains how genetic algorithms can be used to improve the performance of supervised machine learning models by tuning the hyperparameters of the models, either by applying a genetic algorithm-driven grid search, or by using a direct genetic search.

Chapter 9, *Architecture Optimization of Deep Learning Networks*, focuses on artificial neural networks, and discovers how genetic algorithms can be used to improve the performance of neural-based models by optimizing their network architecture. You will then learn how to combine network architecture optimization with hyperparameter tuning.

Chapter 10, *Reinforcement Learning with Genetic Algorithms*, covers reinforcement learning, and explains how genetic algorithms can be applied to reinforcement learning tasks while solving two benchmark environments—MountainCar and CartPole—from the OpenAI Gym toolkit.

Chapter 11, *Genetic Image Reconstruction*, experiments with the reconstruction of a well-known image using a set of semi-transparent polygons, orchestrated by genetic algorithms. Along the way, you will gain useful experience in image processing and the relevant Python libraries.

Chapter 12, *Other Evolutionary and Bio-Inspired Computation Techniques*, broadens your horizons and gets you acquainted with several other biologically inspired problem-solving techniques. Two of these methods—genetic programming and particle swarm optimization—will be demonstrated using DEAP-based Python programs.

To get the most out of this book

To get the most out of this book, you should have a working knowledge of the Python programming language, and basic knowledge of mathematics and computer science. An understanding of fundamental machine learning concepts will be beneficial, but not mandatory, as the book covers the necessary concepts in a nutshell.

To run the programming examples accompanying this book, you will need Python release 3.7 or newer, as well as several Python packages described throughout the book. A Python IDE (Integrated Development Environment), such as PyCharm or Visual Studio Code, is recommended but not required.

Download the example code files

You can download the example code files for this book from your account at www.packt.com. If you purchased this book elsewhere, you can visit www.packtpub.com/support and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at www.packt.com.
2. Select the **Support** tab.
3. Click on **Code Downloads**.
4. Enter the name of the book in the **Search** box and follow the onscreen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of:

- WinRAR/7-Zip for Windows
- Zipeg/iZip/UnRarX for Mac
- 7-Zip/PeaZip for Linux

The code bundle for the book is also hosted on GitHub at <https://github.com/PacktPublishing/Hands-On-Genetic-Algorithms-with-Python>. In case there's an update to the code, it will be updated on the existing GitHub repository.

We also have other code bundles from our rich catalog of books and videos available at <https://github.com/PacktPublishing/>. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: https://static.packt-cdn.com/downloads/9781838557744_ColorImages.pdf.

Code in Action

Visit the following link to check out videos of the code being run:
<http://bit.ly/3azd7Sp>

Conventions used

There are a number of text conventions used throughout this book.

CodeInText: Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. Here is an example: "The `__init__()` method of the class creates the dataset."

A block of code is set as follows:

```
self.X, self.y = datasets.make_friedman1(n_samples=self.numSamples,
                                         n_features=self.numFeatures,
                                         noise=self.NOISE,
                                         random_state=self.randomSeed)
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
self.regressor = GradientBoostingRegressor(random_state=self.randomSeed)
```

Any command-line input or output is written as follows:

```
pip install deap
```

Bold: Indicates a new term, an important word, or words that you see on screen. For example, words in menus or dialog boxes appear in the text like this. Here is an example: "Select **System info** from the **Administration** panel."



Warnings or important notes appear like this.



Tips and tricks appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: If you have questions about any aspect of this book, mention the book title in the subject of your message and email us at customercare@packtpub.com.

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book, we would be grateful if you would report this to us. Please visit www.packtpub.com/support/errata, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at copyright@packt.com with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit authors.packtpub.com.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit packt.com.