# Hyperparameter Optimization in Machine Learning

## Make Your Machine Learning and Deep Learning Models More Efficient

Tanay Agrawal

Apress®

# Hyperparameter Optimization in Machine Learning

## Make Your Machine Learning and Deep Learning Models More Efficient

**Tanay Agrawal**

Apress®

***Hyperparameter Optimization in Machine Learning***

Tanay Agrawal
Bangalore, Karnataka, India

ISBN-13 (pbk): 978-1-4842-6578-9 ISBN-13 (electronic): 978-1-4842-6579-6
https://doi.org/10.1007/978-1-4842-6579-6



Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: Matthew Moodie
Coordinating Editor: Aditee Mirashi

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit https://www.springeronline.com. Apress Media, LLC is a California 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 https://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 via the book's product page, located at https://www.apress.com/978-1-4842-6578-9. For more detailed information, please visit https://www.apress.com/source-code.

Printed on acid-free paper

*This book is dedicated to my parents*
*and my grandparents.*

# Table of Contents

# About the Author

**Tanay Agrawal** is a deep learning engineer and researcher who graduated in 2019 with a bachelor of technology from SMVDU, J&K. He is currently working at Curl Hg on SARA, an OCR platform. He is also advisor to Witooth Dental Services and Technologies. He started his career at MateLabs working on an AutoML Platform, Mateverse. He has worked extensively on hyperparameter optimization. He has also delivered talks on hyperparameter optimization at conferences including PyData, Delhi and PyCon, India.

# About the Technical Reviewer

**Manohar Swamynathan** is a data science practitioner and an avid programmer, with over 14 years of experience in various data science–related areas that include data warehousing, business intelligence (BI), analytical tool development, ad hoc analysis, predictive modeling, data science product development, consulting, formulating strategy, and executing analytics program. He's had a career covering the life cycle of data across different domains such as US mortgage banking, retail/e-commerce, insurance, and industrial IoT. He has a bachelor's degree with a specialization in physics, mathematics, and computers and a master's degree in project management. He's currently living in Bengaluru, the Silicon Valley of India.

# Acknowledgments

I would like to thank Kailash Ahirwar (CTO, MateLabs) for being a great mentor. A big thanks to teams from both MateLabs and Curl HG for their constant support. I am grateful to Akruti Acharya and Jakub Czakon for their insightful inputs while writing this book. I would also like to thank Paankhi Agrawal, Sahil Sharma, Dr. H.-Y. Amanieu, Anubhav Kesari, Abhishek Kumar, Amog Chandrashekar, and others. This book wouldn't have been possible without all the love and encouragement from my family.

# Foreword 1

I have to admit that tweaking parameters by hand was something that I really enjoyed when I trained my first ML models. I would change a parameter, run my training script, and wait to see if the evaluation score improved. One of those guilty pleasures.

But as I spent more time in the ML world, I understood that there are other, more impactful areas where I could spend my time. I realized that I could (and should) outsource parameter tweaking somewhere.

I learned about random search and started using it in my projects. At some point, I felt I could do better than random search and started reading about more advanced hyperparameter optimization algorithms and libraries.

A lot of articles I found where pretty shallow and basic, but I remember reading this deep, hands-on yet easy-to-follow article about Hyperopt, one of the most popular HPO libraries. It was written by Tanay Agrawal. That article probably still is one of the more valuable articles I've ever read on the subject. I mentioned it in one of my blog posts and this is how we met.

When Tanay told me that he was writing a book about hyperparameter optimization, without hesitation, I proposed to review it. I am not going to lie, I really wanted to read it before anyone else! To my surprise, Tanay agreed and I was able to give a few notes here and there.

This book truly is a deep dive into the theory and practice of hyperparameter optimization. I really like how it explains theory deeply but not in an overly complex way. The practical examples are centered on the libraries and frameworks that are heavily used today, which makes this book current and, most importantly, useful.

I recommend this book to any ML practitioner who wants to go beyond the basics and learn the why, how, and what of hyperparameter optimization.

Jakub Czakon
Senior Data Scientist
Neptune.ai

# Foreword 2

In this book, Tanay takes you on an interactive journey—in the most literal sense, as each line of code can be run in a notebook—of the depths of hyperparameters. It helps anyone to quickly get started on tuning and improving their deep learning project with any library they choose to use.

The author mindfully covers the inner workings of hyperparameters in ML models in a thorough but accessible fashion, which will allow you to understand and build upon them using different libraries. The book also demystifies the blackest of the black box: hyperparameter optimization in automated machine learning.

It's a friendly guide to a complicated subject, and yet it's full of cutting-edge gems that even advanced practitioners will love.

Akruti Acharya
Data Scientist
Curl HG

# Introduction

Choosing the right hyperparameters when building a machine learning model is one of the biggest problems faced by data science practitioners. This book is a guide to hyperparameter optimization (HPO). It starts from the very basic definition of *hyperparameter* and takes you all the way to building your own AutoML script using advance HPO techniques. This book is intended for both students and data science professionals.

The book consists of five chapters. Chapter 1 helps you to build an understanding of how hyperparameters affect the overall process of model building. It teaches the importance of HPO. Chapter 2 introduces basic and easy-to-implement HPO methods. Chapter 3 takes you through various techniques to tackle time and memory constraints. Chapters 4 and 5 discuss Bayesian optimization, related libraries, and AutoML.

The intent of this book is for readers to gain an understanding of the HPO concepts in an intuitive as well as practical manner, with code implementation provided for each section. I hope you enjoy it.

CHAPTER 1

# Introduction to Hyperparameters

Artificial intelligence (AI) is suddenly everywhere, transforming everything from business analytics, the healthcare sector, and the automobile industry to various platforms that you may enjoy in your day-to-day life, such as social media, gaming, and the wide spectrum of the entertainment industry. Planning to watch a movie on a video-streaming app but can't decide which movie to watch? With the assistance of AI, you might end up watching one of the recommendations that are based on your past movie selections.

Machine learning is a subset of AI that involves algorithms learning from previous experiences. In some cases, machine learning has achieved human-level accuracy. For example, state-of-the-art deep neural networks (DNNs) perform as well as humans in certain tasks, such as image classification, object detection, and so forth, although this is not the same as simulating human intelligence (but it's a step).

In machine learning algorithms, tuning hyperparameters is one of the important aspects in building efficient models. In this chapter you'll discover the meaning of the term *hyperparameter* and learn how hyperparameters affect the overall process of building machine learning models.


T. Agrawal, *Hyperparameter Optimization in Machine Learning*,
https://doi.org/10.1007/978-1-4842-6579-6_1

## Introduction to Machine Learning

*Machine learning* is the study of algorithms which perform a task without explicitly defining the code to perform it, instead using data to learn. Machine learning enables algorithms to learn on their own without human intervention.

Tom M. Mitchell, a computer scientist and, at the time of writing, a professor at Carnegie Mellon University, defines machine learning as follows: "A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."

Machine learning algorithms have several subdivisions based on the type of problem that needs to be solved. Here I will introduce you to three main types:

- *Supervised machine learning algorithms*: Labeled data is provided, we build a model over it to predict such labels given variables. As an example, suppose you want to purchase a spaceship. Several factors would help you to decide which spaceship to buy: cost, size of spaceship, build quality, whether it has hyperdrive, its weaponry system, and so on. Now we have data of hundreds of spaceships with such feature information and their price, so we build a model and predict the price. This comes under the regression problem. *Regression problems* have continuous target values, and if the target values are discrete, we call them *classification problems*. A third type of problem features time-stamps, *time series forecasting*, where the next data point is somewhat dependent on the previous information, so your algorithm needs to keep in memory information from the previous data points.

The image on the left in Figure 1-1-1 is an example of a classification problem. We need labeled data in order to learn to draw a seperation between them.

- *Unsupervised machine learning algorithms*: These kinds of problems do not have a target value. Suppose you have to group the hypothetical spaceships in clusters according to their features; you would use a *clustering algorithm* to do so. Unsupervised machine learning is used to detect patterns among the dataset. You don't know which cluster is which, but you do know that all the spaceships in one cluster are similar to each other; the right image in Figure 1-1-1 shows an example.

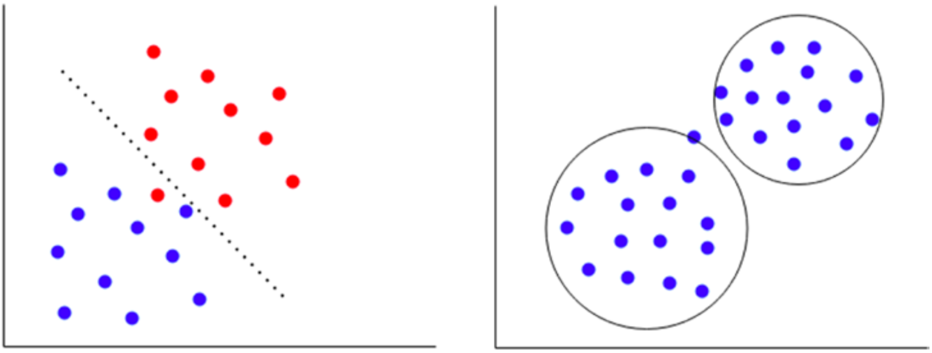

***Figure 1-1-1.*** *Examples of supervised machine learning (left) and unsupervised machine learning (right)*

- *Reinforcement machine learning algorithms*: A reinforcement machine learning algorithm learns from the environment; if it performs well, it gets a reward, and the goal is to maximize the reward. For example, consider the Chrome "running dinosaur" game (go to `chrome://dino/` and press the spacebar). The dinosaur continuously runs toward obstacles. To increase your score, you have to press the spacebar at the precise time to make the dinosaur jump over the obstacles. Here those points are the reward and jumping is the variable that needs to be decided at the right time. In problems like this, we use reinforcement machine learning algorithms. The *Q-learning algorithm* is one example of a reinforcement machine learning algorithm. One of the most brilliant applications of reinforcement machine learning is a robot learning to walk through trial and error.

There's a lot more to machine learning. You need to be familiar with the basics of machine learning before jumping into hyperparameters and their optimization methodologies. If you are new to machine learning or if you want to brush up on the basic concepts, refer to Appendix I and Appendix II. Appendix I covers practical application of machine learning and some of its basic aspects. Appendix II gives you a brief introduction to fully connected neural networks and the PyTorch and Keras frameworks for implementation.

## Understanding Hyperparameters

There are two kinds of variables when dealing with machine learning algorithms, depicted in Figure 1-2-1:

- *Parameters*: These are the parameters that the algorithm tunes according to dataset that is provided (you don't have a say in that tuning)
- *Hyperparameters*: These are the higher-level parameters that you set manually before starting the training, which are based on properties such as the characteristics of the data and the capacity of the algorithm to learn