

DEEP LEARNING IN PRODUCTION



Sergios Karagiannakos

AI SUMMER

DEEP LEARNING IN **PRODUCTION**

Sergios Karagiannakos

AI SUMMER

Table of Contents

Preface	xi
Acknowledgements	xiii
1 About this Book	1
1.1 Welcome to Deep Learning in Production	1
1.2 Is this book for me?	1
1.3 What is the book's goal?	2
1.4 Will this be difficult to learn?	2
1.5 Why should you read this book?	3
1.6 How to use this book?	3
1.7 How is the book structured?	3
1.8 Do I need to know anything else before I get started?	4
2 Designing a Machine Learning System	5
2.1 Machine learning: phase zero	5
2.2 Data engineering	7
2.3 Model engineering	8
2.4 DevOps engineering	8
2.5 Putting it all together	9
2.6 Tackling a real-life problem	9
3 Setting up a Deep Learning Workstation	13
3.1 Laptop setup	14
3.1.1 Laptop requirements	14
3.1.2 Operating system	14
3.2 Frameworks and libraries	14

3.3	Development tools	15
3.3.1	Terminal	16
3.3.2	Version control	16
3.4	Python package and environment management	17
3.4.1	IDE / code editor	18
3.4.2	Other tools	19
4	Writing and Structuring Deep Learning Code	21
4.1	Best practices	21
4.1.1	Project structure	22
4.1.2	Object-oriented programming	23
4.1.3	Configuration	29
4.1.4	Type checking	30
4.1.5	Documentation	31
4.2	Unit testing	33
4.2.1	Basics of unit testing	34
4.2.2	Unit tests in Python	35
4.2.3	Tests in Tensorflow	36
4.2.4	Mocking	37
4.2.5	Test coverage	40
4.2.6	Test example cases	41
4.2.7	Integration / acceptance tests	42
4.3	Debugging	43
4.3.1	How to a debug deep learning project?	43
4.3.2	Python's debugger	44
4.3.3	Debugging data with schema validation	45
4.3.4	Logging	48
4.3.5	Python's Logging module	49
4.3.6	Useful Tensorflow debugging and logging functions	51
5	Data Processing	55
5.1	ETL: Extract, Transform, Load	56
5.2	Data reading	57
5.2.1	Loading from multiple sources	58
5.2.2	Parallel data extraction	59
5.3	Processing	60
5.4	Loading	63
5.4.1	Iterators	64
5.5	Optimizing a data pipeline	65
5.5.1	Batching	66
5.5.2	Prefetching	67

5.5.3	Caching	67
5.5.4	Streaming	69
6	Training	71
6.1	Building a trainer	72
6.1.1	Creating a custom training loop	73
6.1.2	Training checkpoints	76
6.1.3	Saving the trained model	76
6.1.4	Visualizing the training with Tensorboard	77
6.1.5	Model validation	78
6.2	Training in the cloud	79
6.2.1	Getting started with cloud computing	80
6.2.2	Creating a VM instance	82
6.2.3	Connecting to the VM instance	83
6.2.4	Transferring files to the VM instance	83
6.2.5	Running the training remotely	85
6.2.6	Accessing training data from a remote environment	85
6.3	Distributed training	88
6.3.1	Data vs model parallelism	89
6.3.2	Training in a single machine	89
6.3.3	Synchronous training	90
6.3.4	Asynchronous training	95
6.3.5	Model parallelism	97
7	Serving	101
7.1	Preparing the model	101
7.1.1	Building the model's inference function	102
7.2	Creating a web application using Flask	104
7.2.1	Basics of modern web applications	104
7.2.2	Exposing the deep learning model using Flask	105
7.2.3	Creating a client	108
7.3	Serving with uWSGI and Nginx	110
7.3.1	Basic Terminology	110
7.3.2	Designing a serving system	112
7.3.3	Setting up a uWSGI server with Flask	113
7.3.4	Setting up Nginx as a reverse proxy	116
7.4	Serving with model servers	117
7.4.1	Tensorflow Serving vs Flask	118
7.4.2	Export a Tensorflow model	118
7.4.3	Install Tensorflow Serving	119
7.4.4	Load a model	120

7.4.5	Multiple versions support	120
7.4.6	Multiple models support	121
7.4.7	Batching inferences	122
8	Deploying	123
8.1	Containerizing using Docker and Docker Compose	123
8.1.1	What is a container?	124
8.1.2	What is Docker	125
8.1.3	Setting up Docker	125
8.1.4	Building a deep learning Docker image	126
8.1.5	Running a deep learning docker container	131
8.1.6	Creating an Nginx container	132
8.1.7	Defining multi-container Docker apps using Docker Compose .	133
8.2	Deploying in a production environment	136
8.2.1	Using containers in Google Cloud	137
8.2.2	Allowing network traffic to the instance	139
8.2.3	Deploying in Google Cloud	140
8.3	Continuous Integration and Delivery (CI / CD)	141
9	Scaling	145
9.1	A journey from 1 to millions of users	145
9.1.1	First iterations of the machine learning app	147
9.1.2	Vertical vs horizontal scaling	149
9.1.3	Autoscaling	152
9.1.4	Cache mechanisms	152
9.1.5	Monitoring alerts	152
9.1.6	Retraining machine learning models	154
9.1.7	Model A/B testing	156
9.1.8	Offline inference	157
9.2	Growing with Kubernetes	159
9.2.1	What is Kubernetes?	160
9.2.2	Getting started with Kubernetes	161
9.2.3	Deploying with Google Kubernetes Engine	163
9.2.4	Scaling with Kubernetes	169
9.2.5	Updating the application	170
9.2.6	Monitoring the application	170
9.2.7	Running a (re)training job	171
9.2.8	Using Kubernetes with GPUs	173
9.2.9	Model A/B testing	173
10	Building an End-to-End Pipeline	175

10.1	MLOps	175
10.1.1	Basic principles	176
10.1.2	MLOps levels	176
10.2	Building a pipeline using TFX	179
10.2.1	TFX glossary	179
10.2.2	Data ingestion	180
10.2.3	Data validation	181
10.2.4	Feature engineering	182
10.2.5	Train the model	183
10.2.6	Validate model	185
10.2.7	Push model	186
10.2.8	Build a TFX pipeline	186
10.2.9	Run a TFX pipeline	187
10.3	MLOps with Vertex AI and Google Cloud	188
10.3.1	Hands on Vertex AI	190
10.3.2	Experimenting with notebooks	191
10.3.3	Loading data	191
10.3.4	Training the model	194
10.3.5	Deploying to Vertex AI	196
10.3.6	Creating a pipeline	197
10.4	More end-to-end solutions	198
11	Where to Go from Here	201
	Appendix	203
	List of Figures	207
	Index	208
	About the Author	209

Preface

Deep Learning in Production is a product of one year of effort. The pages and the code you will read, begun as articles on our blog AI Summer and they were later combined and organized into a single resource. Some were rewritten from scratch; some were modified to fit the book's structure and some they are entirely new.

The reason I decided to invest the time in writing this book is very simple. The practices and principles mentioned here is what I wish I knew when I started my journey on machine learning. Having an all-in guide that outlines every aspect of the deep learning pipeline would have accelerate my learning curve by a big margin. I do hope that it will do the same for you. Or at least, give you an overview of all the different components and steps, so you can have a holistic view of the field.

It accumulates the knowledge I have gained over the past years by working both as part of the machine learning infrastructure team at HubSpot, as a Data Scientist in a web agency, and as an independent contractor with various start-ups. Each project helped me learn something new and each team gave me a fresh and unique perspective on the field. This is what I hope to transmit to you.

The background and skills that you will acquire from this book will provide you better job opportunities, will differentiate you from other data scientists and machine learning researchers, and the most important thing: they will make you a better and more well-rounded engineer.

Thank you for choosing this book. I deeply appreciate it.

Sergios Karagiannakos

About this Book

In this chapter:

- What this book is about
- What do you need to know beforehand
- Who should read this book

1.1 Welcome to Deep Learning in Production

Hello. It's amazing to have you here. I'm super excited and you should be too. Deep Learning is one of the most transformative technologies of today. Deep neural networks are gradually changing many industries from healthcare and transportation to manufacturing and agriculture. However, the gap between creating a novel neural network architecture and using the model in an actual product is still quite large. The deployment of a machine learning model into production is where the fun begins. This is what this book is all about. **We will discuss the basic steps one should take to build, deploy, scale, and maintain deep learning models.**

1.2 Is this book for me?

Since you are here, I will probably say yes. But let's break it down. If you are interested in learning how to take a simple model and transform it into a real-world

application, then the answer is absolutely yes.

Software engineers

If you come from a software engineering background and you are starting out with deep learning, you will learn how to apply the things you already know on AI projects.

Machine Learning researchers

If you don't have a programming background and you are a deep learning researcher or practitioner, you will familiarize yourself with the software development lifecycle. You will learn the basic principles of how to deploy and scale a machine learning application. In the end, you will be able to create an entirely new deep learning app from scratch and serve it to actual users.

Machine Learning engineers

If you are already a machine/deep learning engineer, you will solidify your understanding of the field, you will explore different practices and ideas, and you will familiarize yourself with software skills and technologies that you might have missed.

Data Scientists

If you are an experienced or aspiring Data Scientist, you will understand what it takes to productionize your models and to build customer-facing applications. You will comprehend the technical debt machine learning has, and you will gain a much broader image of the field.

1.3 What is the book's goal?

If I were to summarize the book into 4 sentences, I'd say that you will learn:

- how to structure and develop production-ready machine learning code
- how to optimize the model's performance and memory requirements
- how to make it available to the public by setting up a service on the cloud
- how to scale and maintain the service as the user base grows

1.4 Will this be difficult to learn?

To be honest with you, it won't be very easy. We cover some advanced concepts and technologies in this book, which might be hard to understand at first. We introduce

a variety of frameworks and libraries that may have a steep learning curve. But don't be discouraged by that. As a machine learning engineer, no one expects you to be an expert on DevOps. However, having a basic understanding of all the systems and concepts, will get you a long way.

1.5 Why should you read this book?

The reason is very simple: because deep learning models are useless if they aren't applied in real-life applications. Don't get me wrong, research is awesome. No one enjoys creating a new fancy, state-of-the-art-model more than me. But the ultimate goal is usually to make the model visible to real users. And that's exactly what you'll learn here.

To clarify why Software Engineering is an undeniable significant part in deep learning, take for example Google Assistant. Behind Google Assistant is, without a doubt, an ingenious machine learning algorithm. But do you think that this amazing research alone is capable of answering the queries of millions of users at the same time? Absolutely not. There are dozens of software engineers behind the scenes, who maintain, optimize, and build the system. This is exactly what we are about to discover.

1.6 How to use this book?

This isn't going to be one of those high-level, abstract books that talk too much without practical value. Here we are going to dive deep into software, we will analyze details that may seem too low level, we will write a lot of code, and we will present the full deep learning development cycle from start to end.

Note, though, that all of the practices mentioned in this book are merely suggestions that come from my experience. Nothing will ever be declared as "the best way to do things".

There are many best practices that, in my opinion, will make your job much easier but that's up to each engineer to decide. After all, this is what makes our job so much fun. The ability to design and build things just the way we want.

Always remember that machine learning systems are not very mature yet, and new solutions and frameworks are created at a rapid pace.

1.7 How is the book structured?

Each chapter of the book will focus on a particular phase of the ML lifecycle.

We will:

1. Start from designing our end system.
2. Explore best practices on how to develop and structure deep learning code.
3. Dive more into the data pre-processing step.
4. Learn how to build optimized data pipelines.
5. Learn how to train models locally or in the cloud.
6. Build a first version of our deployable service.
7. Deploy it in a cloud instance.
8. Discover how to scale and maintain the system.
9. Close with existing end-to-end solutions and efforts to standardize the entire lifecycle.

It's going to be a long but amazing journey.

1.8 Do I need to know anything else before I get started?

Needless to say, that in order to follow along, you will need to have a solid background on programming and/or machine learning.

We will use Tensorflow 2.0 in order to showcase the different techniques and ideas. So, a good understanding of Tensorflow would be ideal. However, all the principles apply to other frameworks such as Pytorch, MXNet and CNTK.

Besides Tensorflow, here is a list of other frameworks and tools that will use throughout the book: Flask, uWSGI, Nginx, Docker, Kubernetes, Google Cloud. As you have probably guessed, this book is quite opinionated in terms of libraries. You can find links with the documentation of all mentioned libraries in the Appendix.

But make no mistake, the choice of the particular tools is only to showcase the concepts. In your own projects, you are of course free to use whatever you like. Along the process, I will also give you some extra tips to better utilize the tools, increase your productivity and enhance your workflow.

Without further ado, let's get started.

Index

A/B Testing, 156
Abstraction, 25
Acceptance tests, 42
API, 102
Autoscaling, 152

Batching, 66

Cache, 152
Checkpoint, 76
CI / CD, 141
Class method, 28
Computational graph, 51
Configuration, 29
Container, 124

Data parallelism, 90
Data processing, 60
Debug, 43
Decorator, 39
Deployment, 123
Docker, 123
Docker compose, 133
Dockerfile, 129
Documentation, 31

ETL, 56
Flask, 105

Functional programming, 60

Horizontal scaling, 150
HTTP, 104

Image segmentation, 10
Inheritance, 26
Iterator, 64

Kubernetes, 160

Linting, 31
Load balancer, 150
Logs, 48

Map-reduce, 156
MLOps, 175
Mocking, 37
Model parallelism, 97
Model server, 117
Monitoring, 152, 170

Nginx, 111

Object storage, 85
Object-oriented programming,
 23
Offline inference, 157
Operating system, 14
Orchestrator, 179

Parallelization, 59
Pipeline, 179
Polymorphism, 28
Prefetching, 67
Python module, 22
Python package, 22

Queue, 157

Retraining, 154
Reverse proxy, 112

Scalability, 146
Schema validation, 45
Static method, 28
Streaming, 69

TensorFlow Extended, 179
Tensorflow Serving, 118

Terminal, 16
Test coverage, 40
Training, 72
Type hints, 30

UNet, 10
Unit tests, 33
uWSGI, 110

Validation, 78
Version control, 16
Vertex AI, 188
Vertical scaling, 149
Virtual environment, 17
Virtual machine, 80

Web service, 102
Web socket, 112

About the Author

Sergios Karagiannakos is a Machine Learning Engineer with a focus on ML infrastructure and MLOps.

He is based in Athens, Greece and graduated with a Master's in Electrical and Computer Engineering from University of Patras. He has helped several companies build their Artificial Intelligence products; most notably, Eworx SA where he joined as a Data Scientist, and Hubspot as part of the Machine Learning infrastructure team.

In 2019, he founded AI Summer, an educational platform around Deep Learning. During this time, he has authored more than 50 articles and he published the Introduction to Deep Learning & Neural Networks course.

Interesting facts: He was included in the Top 100 influential voices and brands in Data Science and Deep Learning, he strives to bring the entire Greek tech community together, and he really wishes that Artificial General Intelligence will be solved in our lifetime.