



Deep Learning Projects Using TensorFlow 2

Neural Network Development with
Python and Keras

Vinita Silaparasetty

Apress®

Deep Learning Projects Using TensorFlow 2

**Neural Network Development
with Python and Keras**

Vinita Silaparasetty

Apress®

Deep Learning Projects Using TensorFlow 2: Neural Network Development with Python and Keras

Vinita Silaparasetty
Bangalore, India

ISBN-13 (pbk): 978-1-4842-5801-9 ISBN-13 (electronic): ISBN 978-1-4842-5802-6
<https://doi.org/10.1007/978-1-4842-5802-6>

Copyright © 2020 by Vinita Silaparasetty

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: Aaron Black
Development Editor: James Markham
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. 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 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 rights@apress.com, or visit <http://www.apress.com/rights-permissions>.

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 via the book's product page, located at www.apress.com/978-1-4842-5801-9. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

Table of Contents

About the Authorxv

About the Technical Reviewerxvii

Acknowledgmentsxix

Prefacexxi

Chapter 1: Getting Started: Installation and Troubleshooting.....1

 Installing Python 32

 Method 1: Direct Installation from the Official Python Website2

 Troubleshooting Tips.....11

 Method 2: Using Anaconda.....12

 Troubleshooting Tips.....24

 Installing Jupyter Notebook26

 Dependencies.....26

 Method 1: Using the PIP Installation Package26

 Troubleshooting Tips.....29

 Method 2: Using Anaconda29

 Troubleshooting Tips.....30

 Installing TensorFlow 2.031

 Dependencies.....31

 Method 1: Using the PIP Installation Package31

 Troubleshooting Tips.....32

 Method 2: Using Anaconda.....32

 Troubleshooting Tips.....34

TABLE OF CONTENTS

Installing Keras34

 Dependencies.....34

 Using the PIP Installation Package34

 Troubleshooting Tips.....36

Installing Python Libraries36

 Installing NumPy.....36

 Installing SciPy38

 Installing Matplotlib40

 Installing Pandas42

 Installing Scikit-Learn45

Summary.....47

Chapter 2: Perceptrons49

 Biological Neurons49

 Artificial Neurons50

 Perceptrons.....51

 Perceptron Learning Rule51

 Types of Activation Functions52

 The Sigmoid Activation Function52

 The ReLU Function.....53

 The Softmax Function.....55

 Perceptrons in Action.....56

 Stage 1: Forward Propagation of Inputs56

 Stage 2: Calculation of the Net Input.....57

 Stage 3: Activation Function58

 Stage 4: Backward Propagation58

 Project Description59

 Important Terminology.....60

Required Libraries	61
Procedure	61
Step 1. Import Libraries.....	61
Step 2. Declare Parameters.....	62
Step 3. Declare the Weights and Bias	62
Step 4. Define the Perceptron Function.....	62
Step 5. Define the Loss Function and Optimizer	63
Step 6. Read in the Data.....	63
Step 7. Visualization of Labels.....	63
Step 8. Prepare Inputs.....	64
Step 9. Initialize Variables	65
Step 10. Train the Model.....	65
Step 11. New Values for Weights and Bias.....	66
Step 12. View the Final Loss	66
Step 13. Predicting Using the Trained Model.....	67
Step 14. Evaluate the Model.....	67
Summary.....	68
Chapter 3: Neural Networks	71
What Is a Neural Network?	71
Neural Network Components	72
Advantages of Neural Networks.....	72
Disadvantages of a Neural Networks.....	72
How a Neural Network Works	73
Forward Propagation.....	74
Backward Propagation.....	75
Types of Neural Networks	75
Feedforward Neural Network.....	76

TABLE OF CONTENTS

Convolutional Neural Networks.....	76
Recurrent Neural Network (RNN)	77
Radial Basis Function Neural Network (RBNN)	78
Project Description.....	79
Flattening Data.....	79
About the Dataset	80
Required Libraries.....	81
Neural Network Architecture.....	81
Procedure.....	81
Summary.....	84
References	86
Chapter 4: Sentiment Analysis	87
LSTM Review	87
How an LSTM Works	89
Layers in an LSTM	92
Project Description.....	93
About the Dataset.....	94
Understanding Sentiment Analysis.....	95
Required Libraries.....	104
LSTM Architecture.....	104
Procedure.....	106
Step 1. Import Libraries.....	106
Step 2. Load the Data	107
Step 3. Prepare the Data	108
Step 4. Clean the Data.....	109
Step 5. Structure the Model	111

Step 6. Compile the Model	111
Step 7. Train the Model.....	112
Step 8. Save the Model (Optional)	112
Step 9. Import the Pretrained Model (Optional)	112
Further Tests	112
Troubleshooting	113
Summary.....	115
References	117
Further Reading	117
Chapter 5: Music Generation	119
GRU Overview	119
How a GRU Works	120
GRU Stages.....	121
GRU Layers	124
Comparing GRU and LSTM	125
Project Description.....	126
About the Dataset	127
Important Terminology and Concepts.....	128
Required Libraries	129
Installation Instructions.....	130
Using PIP	130
Using Windows	132
Using macOS	132
Using Linux.....	133
Installation Troubleshooting.....	133
GRU Architecture	134

TABLE OF CONTENTS

Procedure..... 136

 Step 1. Import Libraries..... 136

 Step 2. Load the Data 137

 Step 3. Feature Extraction 137

 Step 4. Exploratory Data Analysis (EDA) 139

 Step 5. Data Preparation (Input) 140

 Step 6. Structure the Model 142

 Step 7. Train the Model..... 144

 Step 8. Prediction 145

 Step 9. Data Preparation (Offset)..... 146

 Step 10. Store the Output as a MIDI File 149

Further Tests 150

Troubleshooting 150

Summary..... 151

References..... 153

Resources 154

Further Reading 154

Chapter 6: Image Colorization 157

 Human Vision Review..... 157

 Computer Vision Review..... 159

 How a CNN Works 161

 Input Layer..... 161

 Convolution Layer: The Kernel 162

 Upsampling Layer..... 163

 DepthwiseConv2D 164

 Pooling Layer..... 164

 Fully Connected Layer 165

Project Description.....	166
About the Dataset.....	167
Important Terminology.....	167
Required Libraries	172
Installation Instructions	173
CNN+VGG-16 Architecture.....	176
Procedure.....	179
Step 1. Import the Libraries.....	179
Step 2. Convert the Images to Grayscale	180
Step 3. Load the Data	181
Step 4. Structure the Model	182
Step 5. Set the Model Parameters.....	183
Step 6. Data Preparation	183
Step 7. Train the Model.....	183
Step 8. Obtain Predictions	184
Step 9. View the Results.....	184
Troubleshooting	185
Further Tests	186
Summary.....	186
References	187
Further Reading	188
Chapter 7: Image Deblurring	189
What Is a GAN?	189
Types of GANs.....	190
How a GAN Works	192
The Generative Model.....	192
Process Within the Generator	193

TABLE OF CONTENTS

The Discriminator Model	194
Process Within the Discriminator	195
Project Description.....	197
About the Dataset	198
Important Terminology and Concepts.....	199
GAN Architecture	201
Required Libraries	202
GAN Architecture.....	203
Generator.....	203
Discriminator	203
Procedure.....	204
Step 1. Import the Libraries.....	204
Step 2. Dataset Preparation.....	205
Step 3. Exploratory Data Analysis.....	207
Step 4. Structure the Model	209
Step 5. Input Preparation.....	215
Step 6. View the Images	217
Step 7. Save Results.....	217
Troubleshooting	218
Further Tests	219
Summary.....	219
References.....	221
Further Reading	221
Chapter 8: Image Manipulation	223
Project Description.....	223
Important Terminology and Concepts.....	224
Copy-Move Forgeries	231

About the Dataset	233
Required Libraries	234
Troubleshooting	235
CNN Architecture	235
Procedure	237
Step 1. Import the Libraries	237
Step 2. Preparing the Dataset	238
Step 3. Structure the Model	242
Step 4. Train the Model	243
Step 5. Test the Model	243
Step 6. Check the Results	244
Further Tests	244
Summary	245
References	247
Further Reading	247
Chapter 9: Neural Network Collection	249
Neural Network Zoo Primer	249
Neural Networks	250
Recurrent Neural Networks (RNNs)	250
Multiplicative LSTM	253
ANNs with Attention	255
Transformers	257
Autoencoder	260
Variational Autoencoders	263
Denoising Autoencoders	266
Recurrent Autoencoders	269
Sparse Autoencoders	271

TABLE OF CONTENTS

Stacked Autoencoders.....	273
Convolutional Autoencoders	275
Stacked Denoising Autoencoders	278
Contractive Autoencoders.....	281
Markov Chains.....	285
Hopfield Networks	286
Bidirectional Associative Memory	289
Boltzmann Machines	290
Restricted Boltzmann Machines.....	292
Deep Belief Networks.....	294
Deconvolutional Networks.....	296
Deep Convolutional Inverse Graphics Networks.....	298
Liquid State Machines	301
Echo State Networks (ESNs).....	303
Deep Residual Network (ResNet)	305
ResNeXt.....	307
Neural Turing Machines.....	312
Capsule Networks	315
LeNet-5.....	317
AlexNet	318
GoogLeNet.....	321
Xception.....	324
Optimizers.....	327
Stochastic Gradient Descent	327
RMSProp.....	329
AdaGrad	331
AdaDelta	332
Adam	333

Adamax.....	335
Nesterov Accelerated Gradient (NAG)	336
Nadam	336
Loss Functions	338
Mean Squared Error (MSE)	338
Mean Absolute Error (MAE).....	338
Mean Absolute Percentage Error (MAPE)	339
Mean Squared Logarithmic Error (MSLE)	339
Squared Hinge	339
Hinge	340
Categorical Hinge	340
Log Cosh.....	340
Huber Loss.....	341
Categorical Cross-Entropy	341
Sparse Categorical Cross-Entropy.....	341
Binary Cross-Entropy.....	342
Kullback-Leibler Divergence	342
Poisson	342
References	343
Further Reading	344
Appendix: Portfolio Tips.....	349
Data Analyst Portfolios	349
Sharing Your Portfolio	353
Types of Projects.....	354
Defining Problem Statements	361
Using Design Thinking	365
Solution Implementation	372

TABLE OF CONTENTS

Maintenance392

Tips for Documenting Projects.....394

Appendix Checklist395

References402

Further Reading402

Resources for Building Your Portfolio.....403

Read.me Template405

Data Cleaning Checklist.....408

Index.....411

About the Author



Vinita Silaparasetty is a data science trainer who is passionate about AI, machine learning, and deep learning. She is experienced in programming with Python, R, TensorFlow, and Keras.

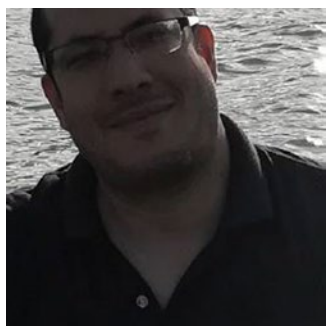
She is currently pursuing her master's degree in Data Science at NewCastle University, U.K.

She has written two award-winning research papers on machine learning. The first is titled “Python vs. R” and is a comparative study of Python and R. The second is titled “Machine Learning for Fraud Detection: Employing Artificial Intelligence in the Banking Sector” and it proposes a new system for real-time fraud detection in the banking sector. She is also a reviewer for the Oxford Publication entitled *The Computer Journal*.

She is the co-organizer of the “Bangalore Artificial Intelligence Meetup” group as well as the “AI for Women” meetup group, where she conducts training sessions on Python, R, machine learning, and deep learning.

You can find her on Facebook at <https://www.facebook.com/VinitaSilaparasetty/>

About the Technical Reviewer



Mezgani Ali is a Ph.D. student in artificial intelligence (Mohamed V. University in Rabat) and a researcher at Native LABS, Inc. He likes technology, reading, and his little daughter Ghita. His first program was a Horoscope in Basic in 1993. He has done a lot of work on the infrastructure side in system engineering, software engineering, managed networks, and security.

Mezgani worked for NIC France, Capgemini, and HP, and he was part of the (SRE) Site Reliability Engineer team responsible for keeping data center servers and customers' applications up and running. He is fanatical about Kubernetes, REST API, MySQL, and Scala, and he is the creator of the functional and imperative programming language, PASP.

Acknowledgments

I would like to thank my father, Mr. S. Mohan Kumar, for his guidance and valuable input that made this book a great resource for beginners and seasoned professionals alike.

I would like to thank my mother, Mrs. Agnes Shanthi Mohan, for her constant support, encouragement, and love.

I would like to thank my younger sister, Ms. Nikita Silaparasetty, for her valuable feedback and support.

Special thanks to Aaron Black, the senior editor, for accepting my book proposal and giving me the opportunity to write this book.

Thanks to Jessica Vakili, the coordinating editor, for ensuring that the process of writing this book was smooth and for clarifying even the smallest doubts I had as a first-time author.

Thanks to James Markham, the editor, for his guidance on formatting each chapter as well as his keen eye for detail, which make this book easy to understand.

Thanks to Mezgani Ali, the technical reviewer, for ensuring that the source code is well formatted.

Finally, I would like to thank the awesome team at Apress, for their effort in making this book possible.

Preface

TensorFlow 2.0 was officially released on September 30th, 2019. However, the new version is very different than what most users are familiar with. While programming with TensorFlow 2.0 is much simpler, most users still prefer to use older versions. This book aims to help long-time users of TensorFlow adjust to TensorFlow 2.0 and to help absolute beginners learn TensorFlow 2.0.

Why use TensorFlow?

Here are some advantages to using TensorFlow for your deep learning projects.

- It is open source.
- It is reliable (has minimal major bugs).
- It is ideal for perceptual and language understanding tasks.
- It is capable of running on CPUs and GPUs.
- It is easier to debug.
- It uses graphs for numeric computations.
- It has better scalability, as libraries can be deployed on a gamut of hardware machines, starting from cellular devices to computers with complex setups.

PREFACE

- It has convenient pipelining, as it is highly parallel and designed to use various backend software (GPU, ASIC, etc.).
- It uses the high-level Keras API.
- It has better compatibility.
- It uses TensorFlow Extended (TFX) for a full production ML pipeline.
- It also supports an ecosystem of powerful add-on libraries and models to experiment with, including Ragged Tensors, TensorFlow Probability, Tensor2Tensor, and BERT.

TensorFlow 1.x	TensorFlow 2.0
Global variables	decorator <code>tf.function</code> so that the following function block is run as a single graph.
Sessions	Functions
Layers	Keras Layers
Placeholders	Eager Execution
<code>tf.app</code> , <code>tf.flags</code> , <code>tf.logging</code>	<code>absl-py</code>
-	The <code>tf.function()</code> will create a separate graph for every unique set of input shapes and datatypes.

Figure I-1. Comparison of TensorFlow 1.x and TensorFlow 2.0

About the Book Projects

The projects in this book mainly cover image and sound data. They are designed to be as simple as possible to help you understand how each neural network works. Consider them to be a skeletal structure for your own projects. You are encouraged to build on the models in this book and experiment with them using different datasets. The projects in this book were designed keeping in mind the latest developments in deep learning and will be the perfect addition for an impressive data science portfolio.

System Specifications

The projects in this book require powerful computing resources or a good cloud platform. You are strongly advised to use a system with the following minimum requirements:

GPU: Model: 16-bit Memory: 8GB and CUDA
Toolkit support

RAM: Memory: 10GB

CPU: PCIe lanes: 8 Core: 4 threads per GPU

SSD: Form Factor: 2.5-inch and SATA interface

PSU: 16.8 watts

Motherboard: PCIe lanes: 8

If you are unable to acquire a system with these requirements, try using a cloud computing platform, such as one of the following:

- BigML
- Amazon Web Services
- Microsoft Azure
- Google Cloud

PREFACE

- Alibaba Cloud
- Kubernetes

Tips to Get the Most Out of This Book

To get the most value out of the projects in this book, follow these guidelines:

- **Create separate environments.** To prevent problems, it's best to create separate environments for each project. This way you will have only the libraries necessary for that particular project and there will not be any clashes.
- **Save your projects in separate folders.** To keep your work organized and handy for future reference, create separate folders for each project. You can store the script, datasets, and results that you have obtained in that folder. Each project in this book provides the code to set your file path to work directly in the project folder that you created.
- **Use data wisely.** Ensure that you have enough data to divide into training and test sets. I suggest that you use 80% of the data for training and 20% for testing.
- **Be organized.** By creating a folder for your project, you know that all the data, output files, etc. are available in one place.
- **Make backups.** Make copies of each notebook before experimenting. This way you have one working copy as a template for future projects. Then make copies of it and modify it as required.

- **Plan.** Understand the problem statement and create a rough flowchart of your approach to solving the problem.
- **Consider your presentation.** As a data scientist, your inferences will be discussed by members of a company who have technical knowledge as well as those who do not. So be sure that you can convey your findings in a manner that anyone can understand.
- **Network.** Join online communities where you can ask questions and help others with solutions to their questions. This is the best way to learn. I recommend the following:
 - StackOverflow
 - Quora
 - Reddit
 - StackExchange
 - CodeProject
 - Google Groups
 - CodeRanch
 - Programmers Heaven
- **Practice:** Need inspiration for more projects? Join online communities that have hackathons, competitions, etc., to help you practice and learn. I recommend the following:
 - Hackerearth
 - Kaggle
 - Challengerocket
 - Angel Hack

CHAPTER 1

Getting Started: Installation and Troubleshooting

In order to make the best use of this book, you'll need to satisfy the following prerequisites:

- Install Python 3, the latest version of Python
- Install Jupyter Notebooks
- Install TensorFlow 2.0
- Install Keras
- Install NumPy
- Install SciPy
- Install Matplotlib
- Install Pandas
- Install Scikit-Learn

This chapter will help you install all the necessary packages. It also provides troubleshooting tips for some common errors that may occur during installation.

Note It is good practice to create a separate virtual environment for these projects. Before installing the packages mentioned here, create a virtual environment and activate it.

Installing Python 3

Python is a general-purpose interpreted, imperative, object-oriented, high-level programming language. It is one of the oldest programming languages around. However, with the onset of machine learning, Python has been given a new lease on life. It has become a popular tool for both machine learning and deep learning. Currently, Python is available as two distinct versions—Python 2 and Python 3.

All the projects in this book use Python 3, so it is best to ensure that it is installed.

Method 1: Direct Installation from the Official Python Website

This method works well with Windows, Linux, and macOS X systems. It is the standard method of installation, whereby you download Python directly from the official website and then install it on your system.

1. Go to <https://www.python.org/> and select the Downloads tab. A drop-down menu will appear (see Figure 1-1).

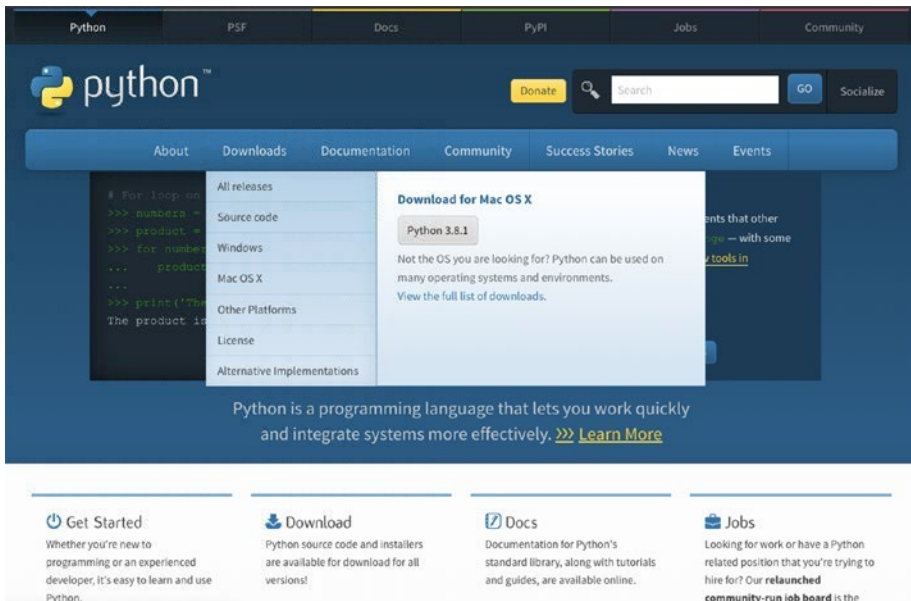


Figure 1-1. Official Python website

2. To the right of the drop-down menu, you will see the latest versions of Python that are available for your specific system. The first button provides the latest version of Python 3. Once you click it, the download will begin.
3. Once the download is complete, double-click the package in the Download bar. This will start the installation process.

4. In the dialog box that pops up, select Continue (see Figure 1-2).



Figure 1-2. Introduction window of the Python installation

5. In the new dialog box, you will be presented with important information regarding the changes made to Python (see Figure 1-3). Once again, select Continue.



Figure 1-3. *The Read Me window of the Python installation*

6. Now you will be shown the license agreement for using Python. Select Continue (see Figure 1-4).



Figure 1-4. The License window of the Python installation

7. A mini dialog box will appear requesting you to agree to the terms and conditions listed. Select Agree.
8. Select the file path for the new Python installation.
9. Select the type of installation. See Figure 1-5.

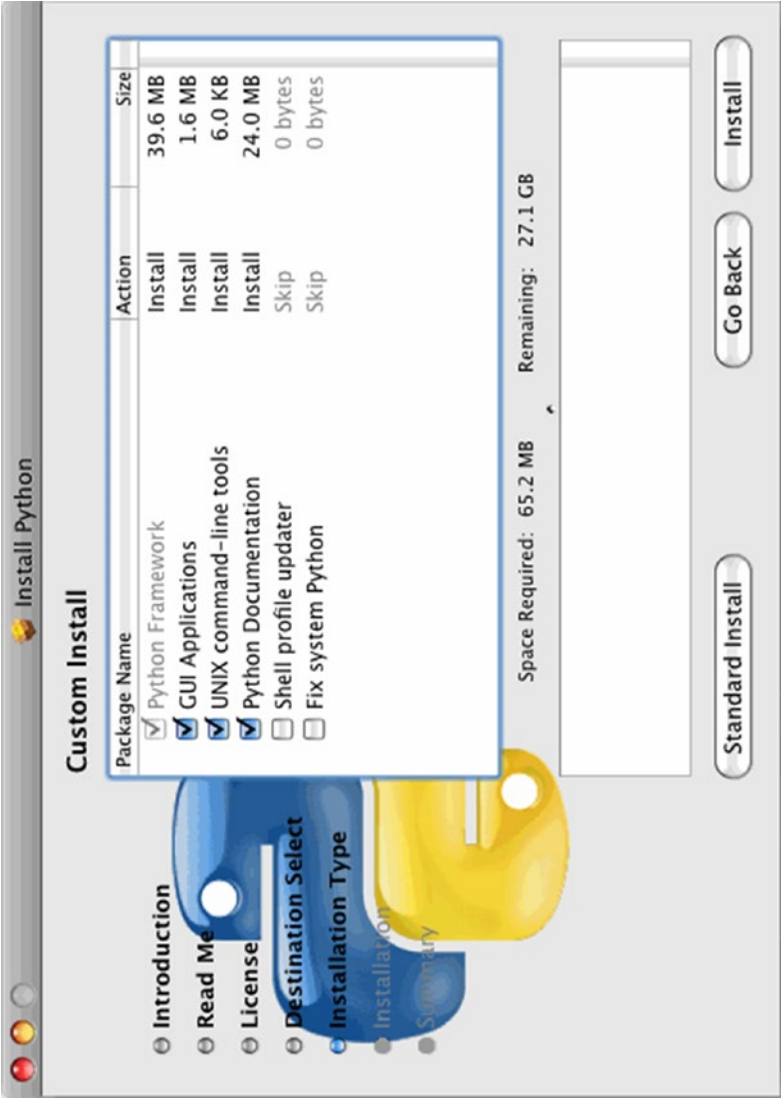


Figure 1-5. The Installation type window of the Python installation