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

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 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

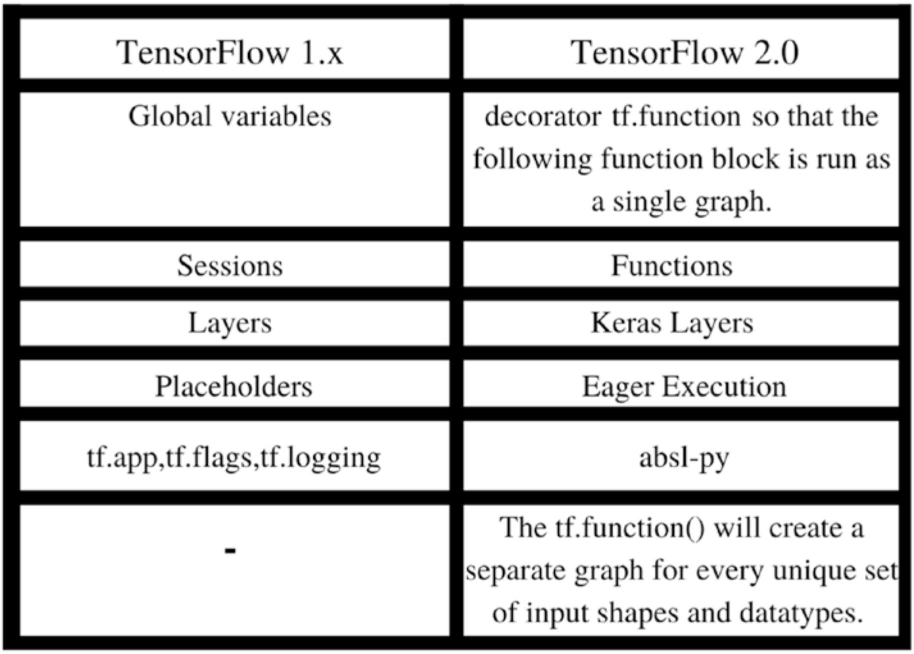
TensorFlow 2.0 was officially released on September 30th, 2019. However, the new version is very different than what most users are familiar with. While programming with TensorFlow 2.0 is much simpler, most users still prefer to use older versions. This book aims to help long-time users of TensorFlow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

- 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 tf.function so that the following function block is run as a single graph.
Sessions	Functions
Layers	Keras Layers
Placeholders	Eager Execution
tf.app,tf.flags,tf.logging	absl-py
-	The tf.function() 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

- 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.

V. Silaparasetty, *Deep Learning Projects Using TensorFlow 2*,
https://doi.org/10.1007/978-1-4842-5802-6_1

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

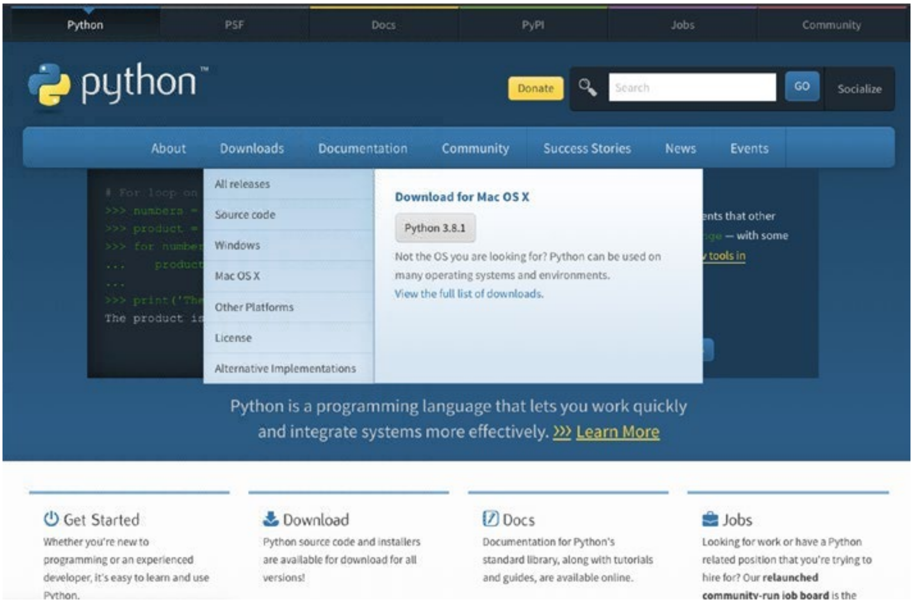
Python is a general-purpose interpreted, imperative, object-oriented, high-level programming language. It is one of the oldest programming languages around. However, with the onset of machine learning, Python has been given a new lease on life. It has become a popular tool for both machine learning and 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).



Not the OS you are looking for? Python can be used on many operating systems and environments. View the full list of downloads.

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> Learn More

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Python source code and installers are available for download for all versions!

Documentation for Python's standard library, along with tutorials and guides, are available online.

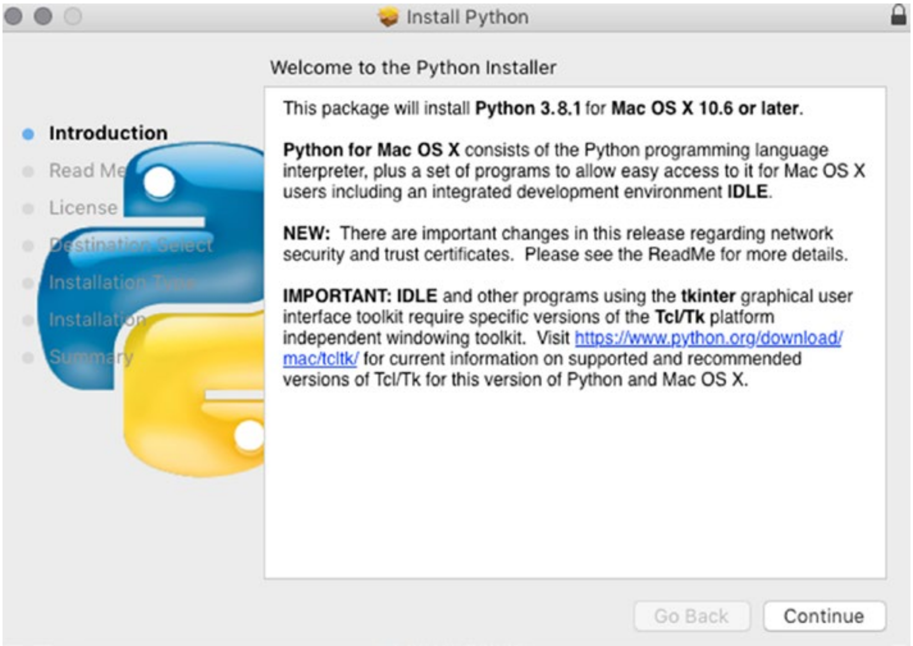
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the

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).



Welcome to the Python Installer

This package will install **Python 3.8.1** for **Mac OS X 10.6 or later**.

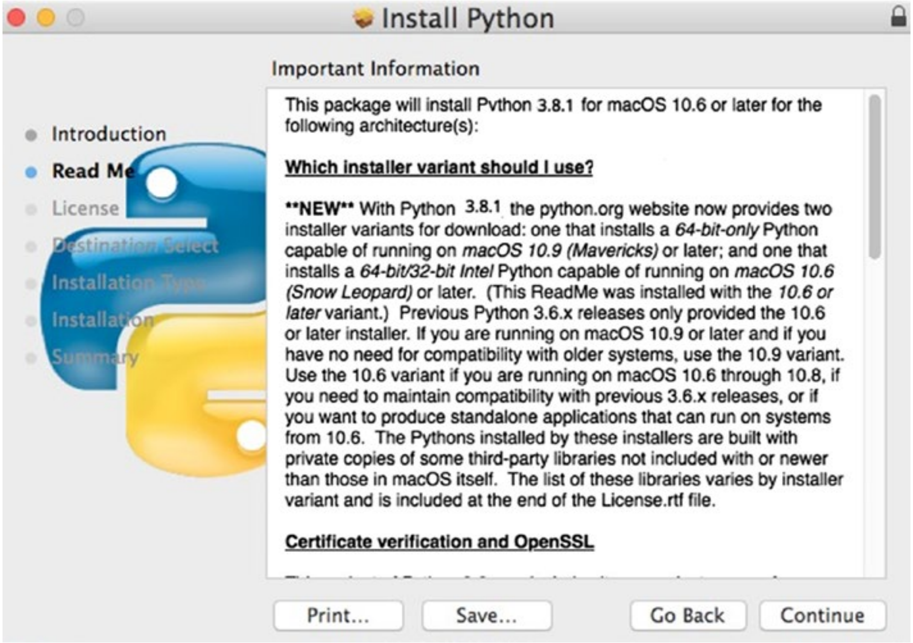
Python for Mac OS X consists of the Python programming language interpreter, plus a set of programs to allow easy access to it for Mac OS X users including an integrated development environment **IDLE**.

NEW: There are important changes in this release regarding network security and trust certificates. Please see the ReadMe for more details.

IMPORTANT: IDLE and other programs using the **tkinter** graphical user interface toolkit require specific versions of the **Tcl/Tk** platform independent windowing toolkit. Visit https://www.python.org/download/mac/tcltk/ for current information on supported and recommended versions of Tcl/Tk for this version of Python and Mac OS X.

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.



Important Information

This package will install Pvthon 3.8.1 for macOS 10.6 or later for the following architecture(s):

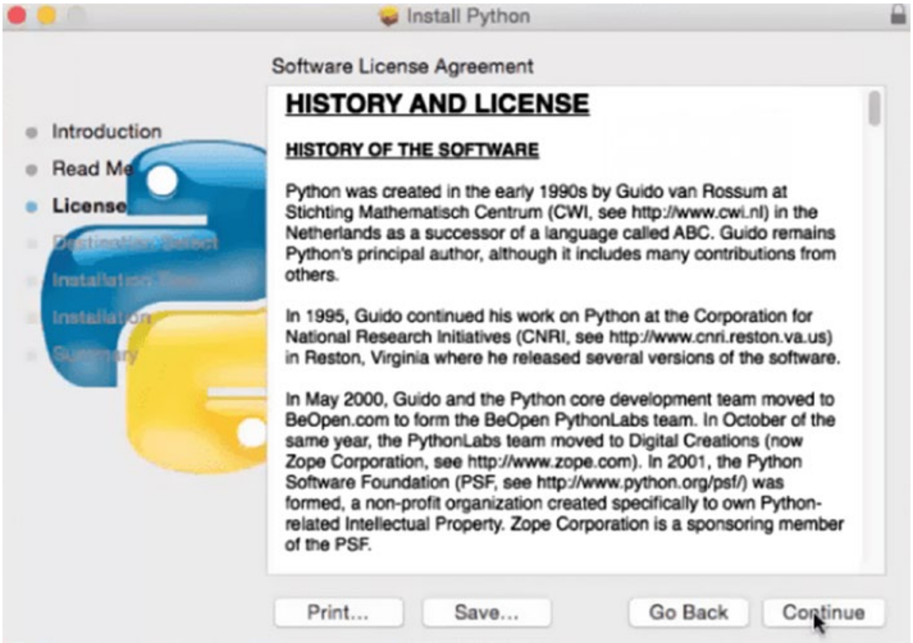
Which installer variant should I use?

NEW With Python 3.8.1 the python.org website now provides two installer variants for download: one that installs a *64-bit-only* Python capable of running on *macOS 10.9 (Mavericks)* or later; and one that installs a *64-bit/32-bit Intel* Python capable of running on *macOS 10.6 (Snow Leopard)* or later. (This ReadMe was installed with the *10.6 or later* variant.) Previous Python 3.6.x releases only provided the 10.6 or later installer. If you are running on macOS 10.9 or later and if you have no need for compatibility with older systems, use the 10.9 variant. Use the 10.6 variant if you are running on macOS 10.6 through 10.8, if you need to maintain compatibility with previous 3.6.x releases, or if you want to produce standalone applications that can run on systems from 10.6. The Pythons installed by these installers are built with private copies of some third-party libraries not included with or newer than those in macOS itself. The list of these libraries varies by installer variant and is included at the end of the License.rtf file.

Certificate verification and OpenSSL

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).



HISTORY AND LICENSE

HISTORY OF THE SOFTWARE

Python was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum (CWI, see http://www.cwi.nl) in the Netherlands as a successor of a language called ABC. Guido remains Python's principal author, although it includes many contributions from others.

In 1995, Guido continued his work on Python at the Corporation for National Research Initiatives (CNRI, see http://www.cnri.reston.va.us) in Reston, Virginia where he released several versions of the software.

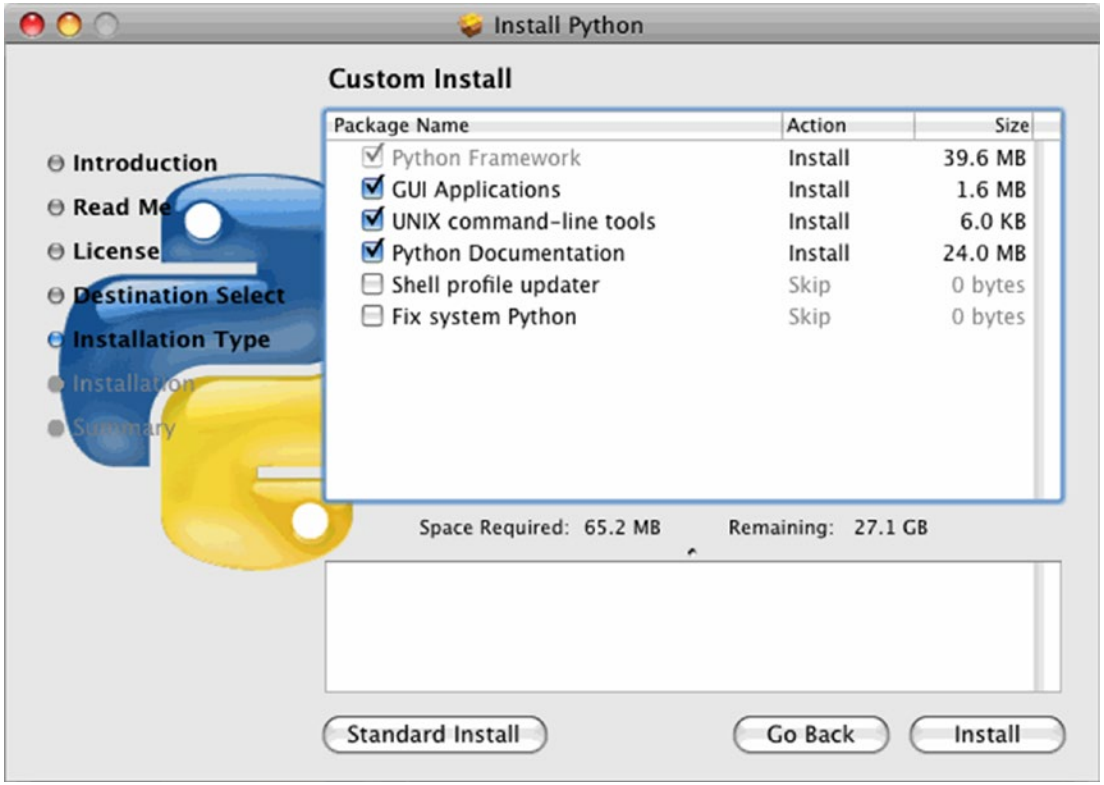
In May 2000, Guido and the Python core development team moved to BeOpen.com to form the BeOpen PythonLabs team. In October of the same year, the PythonLabs team moved to Digital Creations (now Zope Corporation, see http://www.zope.com). In 2001, the Python Software Foundation (PSF, see http://www.python.org/psf/) was formed, a non-profit organization created specifically to own Python-related Intellectual Property. Zope Corporation is a sponsoring member of the PSF.

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.



Custom Install

Package Name	Action	Size
☑ Python Framework	Install	39.6 MB
☑ GUI Applications	Install	1.6 MB
☑ UNIX command-line tools	Install	6.0 KB
☑ Python Documentation	Install	24.0 MB
□ Shell profile updater	Skip	0 bytes
□ Fix system Python	Skip	0 bytes

Space Required: 65.2 MB Remaining: 27.1 GB

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