Manual for the Installation of Python

Jennifer Wendland and Prof. Dr. Maik Kschischo

April 9, 2025

Introduction

To internalize the lecture material of the **Deep Learning** module, we will conduct exercises throughout the semester using the programming language Python. In this course, we offer two options for using Python (see also `https://d2l.ai/chapter_installation/index.html`):

A) Use Google Colab Python online – this utilizes computational resources provided by Google.

B) Use Anaconda Python, an open-source distribution, locally on your own computer.

A) Google Colab

A prerequisite for using Google Colab is creating a Google account, including Google Drive (see also `https://d2l.ai/chapter_installation/index.html`). The advantages of Google Colab include the provision of computational resources by Google for running scripts and the pre-installation of many packages. In Google Colab, it is not (readily) possible to create an environment – this step is thus omitted when setting up Google Colab. Here, one must work directly with "notebooks" – similar to Jupyter Notebooks – where all packages (except d2l) that we need are already pre-installed.

First Steps

Under "File - New Notebook," new scripts can be created. At the beginning of a script, the d2l package must be installed with "pip install d2l" (which takes about 4 seconds). Files stored on Google Drive can be found under Files. For safety, I recommend downloading the created notebook by selecting "File - Download as - .ipynb" and saving it locally.

B) Anaconda

Anaconda, as an open-source distribution for Python, offers several advantages over the plain version of Python. Anaconda is universally usable on Linux, Windows, and Mac and includes, among other things, the development environments Spyder and Jupyter Notebook. Installation A detailed explanation for installing Anaconda on Windows, Linux, and Mac can be found at the following link: `https://docs.anaconda.com/anaconda/install/index.html`.

Virtual Environments with Anaconda

A Virtual Environment is an environment where packages of a specific project can be installed into a path other than the default path. It often happens that certain code (or in general: projects) only runs with specific package versions, or some packages are not compatible with each other (for example, TensorFlow 1 code is not easily usable with TensorFlow 2). When multiple projects with different package dependencies need to be managed on a single computer, a sensible solution is to create a Virtual Environment, allowing the installation of the required packages into the environment of each respective project. Here are a few basic terms:

- Creation of a Virtual Environment: conda create –name myenv
- Activation of a Virtual Environment: conda activate myenv
- Deactivation of a Virtual Environment: conda deactivate

Additional information about Virtual Environments can be found at the following link:
`https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#deactivatingan-environment`
We create an environment for our lecture in the following way:
conda create –name d2l python=3.9 -y

Installation of Packages

For the Deep Learning course, the following packages are required: IMPORTANT: Activate the Virtual Environment before installing the packages!

- d2l
- pytorch
- (spyder)

The following packages were previously automatically installed as dependencies

- Jupyterlab
- numpy
- matplotlib