Local Installation Guide

I. Install Python

The very first step is the installation of python on your local machine. The easiest way is to use the Anaconda Python distribution to install most of the Python packages we will need:

https://www.anaconda.com/products/individual

You can choose from Windows, Mac and Linux OS as per your requirement. After selecting the OS and Python version, it will download the Anaconda installer on your computer. Now, double click the file and the installer will install it for you.

II. Jupyter Notebook

We will use Colab for the lecture and the lecture notes will be shared as Jupyter notebooks. We will use it as it provides an interactive computational environment for developing Python based applications and it also helps us to document the work as we progress.

If you are using Anaconda distribution, then you do not need to install the jupyter notebook separately as it is already installed with it. You just need to go to Anaconda Prompt and type the following command

```
jupyter notebook
```

It will provide you almost the same interface we use in the lecture.

if you are using standard Python distribution then jupyter notebook can be installed using popular python package installer, pip:

```
pip install jupyter
```

III. NumPy and SciPy

NumPy and SciPy are open-source add-on modules to Python that provides commonly used mathematical and numerical routines. If you are using Anaconda distribution, then no need to install them separately as they are already installed with it. Otherwise:

```
pip install NumPy
pip install scipy
```

https://numpy.org/

https://www.scipy.org/

IV. Pandas

Pandas is basically used for data manipulation, wrangling and analysis. It was developed by Wes McKinney in 2008. If you are using Anaconda distribution, then no need to install Pandas separately as well. Otherwise:

```
pip install Pandas
```

https://pandas.pydata.org/

V. Matplotlib and Seaborn

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. If you are manually installing all packages:

```
pip install -U matplotlib
```

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics:

```
pip install seaborn
```

The library is also included as part of the Anaconda distribution:

```
conda install seaborn
```

https://matplotlib.org/
https://seaborn.pydata.org/

VI. Scikit-learn

This is the ML library we will use during the fundamentals. If you are using Anaconda distribution, then no need to install Scikit-learn separately as it is already installed with it. You just need to use the package into your Python script. Otherwise:

```
pip install -U scikit-learn
```

➢ Hint:

You can also install them all at once. For example:

```
python3 -m pip install -U jupyter matplotlib numpy pandas scipy scikit-learn
```

or

```
conda install numpy pandas scikit-learn matplotlib seaborn scipy
```

VII. TensorFlow and Keras

you will need to install TensorFlow and Keras to build neutral networks after the fundamentals section.

On Windows open the Start menu and open an Anaconda Command Prompt. On macOS or Linux open a terminal window. Use the default bash shell on macOS or Linux. Then choose a name for your TensorFlow environment, such as “tf”.

To install the current release of CPU-only TensorFlow, recommended for beginners:

```
conda create -n tf tensorflow
conda activate tf
```

Or, to install the current release of GPU TensorFlow on Linux or Windows:

```
conda create -n tf-gpu tensorflow-gpu
conda activate tf-gpu
```

TensorFlow is now installed and ready to use.

For using TensorFlow with a GPU, refer to the TensorFlow documentation on the topic, specifically the section on device placement. As far as I know, it works well on NVIDIA but you have to put some effort to operate on other brands.

GPU TensorFlow uses CUDA. On Windows and Linux only CUDA 10.0 is supported for the TensorFlow 2.0 release. Previous versions of TensorFlow support other version of CUDA.

To install GPU TensorFlow with a non-default CUDA version such as 9.0 run:

```
conda create -n tf-gpu-cuda8 tensorflow-gpu cudatoolkit=9.0
conda activate tf-gpu-cuda8
```

TensorFlow 2 packages require a `pip` version >19.0. For pip:

```
# Requires the latest pip
pip install --upgrade pip

# Current stable release for CPU and GPU
pip install tensorflow
```

Or

```
python3 -m pip install -U tensorflow
```

Note: conda installation is recommended for higher performance!

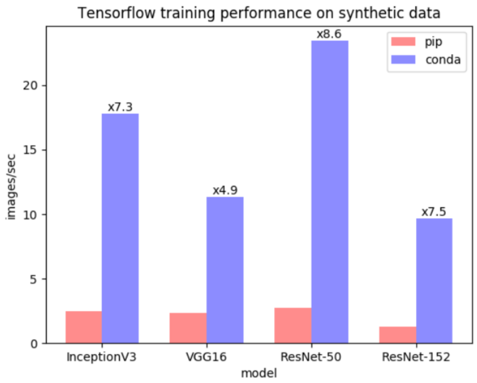

Figure 1: Training performance of TensorFlow on a number of common deep learning models using synthetic data. Benchmarks were performed on an Intel® Xeon® Gold 6130.

See https://www.anaconda.com/blog/tensorflow-in-anaconda

VIII. Additional Resources

https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks