Tutorial 0 - Machine Setup

Instructions: This assignment is ungraded. Please use it to setup your machines.

1. Introduction

In this tutorial, we will walk you through the process of creating a virtual Python environment and installing essential dependencies for upcoming assignments.

2. What is a Virtual Environment?

A virtual environment is a self-contained directory that contains a Python installation for a particular version of Python, along with a number of additional packages. For more information, please refer to Python tutorial.

3. Prerequisites

Before we start, ensure you have Python installed on your computer.

If you don't have Python installed, please follow the following steps.

3.1 Download Python from the Official Website

You can download Python from the official Python website.

Choose the version of Python you want to install (preferably the latest stable release). Click on the download button, and the installation file for your operating system (Windows or macOS) will begin downloading.

3.2 Installation on Different Operating Systems

Windows

1. **Run the Installer**: Once the download is complete, double-click the installer to run it.

2. **Important: Check the "Add Python to PATH" Option**: Before clicking "Install Now", ensure that you **check the box** that says "Add Python to PATH". This step is crucial for allowing you to run Python from the command line without needing to navigate to the Python installation directory.

3. **Choose Installation Options**: Select "Install Now" to install Python with default settings.

4. **Wait for Installation to Complete**: Once the installation is complete, you will see a "Setup was successful" message. Click "Close".

5. **Verify the Installation**: Open a new Command Prompt window (`cmd`) and type:

```
python --version
```

If Python is installed correctly, you'll see the version number displayed.

macOS

1. **Run the Installer**: Open the downloaded `.pkg` file to run the installer.
2. **Follow the Installation Wizard**: The installation process is straightforward. Follow the on-screen instructions to install Python on your system.
3. **Verify the Installation**: Open Terminal and type:

```
python --version
```

If Python is installed correctly, you'll see the version number displayed.

4. **Check PATH**: macOS typically manages Python installations with its own system version of Python. Ensure that the downloaded Python version is being used by typing:

```
which python3
```

in terminal.

This should return the path to the Python installation you just added.

3.3 Installing pip

Most Python installations come with `pip` (Python's package manager) by default. To verify if `pip` is installed, use the following command in terminal:

```
pip --version
```

If it's not installed, use the built-in `ensurepip` module to install `pip`(for more information, please refer to pip documentation):

```
python -m ensurepip --upgrade
```

4. Step-by-Step Guide to Creating a Virtual Environment

4.1 Check Python Installation

First, let's confirm that Python is installed and available in your terminal.

- **Windows**: Open Command Prompt (`cmd`) and type:

```
python --version
```

- **macOS**: Open Terminal and type:

```
python --version
```

If Python is installed, you should see a version number like `Python 3.x.x`.

4.2 Install `venv` (if not already installed)

For most installations of Python 3, `venv` is included by default. You can check if it's available:

- **Windows:** Open Command Prompt (`cmd`) and type:

```
python -m venv --help
```

- **macOS:** Open Terminal and type:

```
python3 -m venv --help
```

If there are no errors, you're good to go. If venv isn't installed, refer to the Python documentation for instructions.

4.3 Create a Virtual Environment

Now, let's create a virtual environment. Choose a directory for your project, and in that directory, run:

- **Windows**: Open Command Prompt (cmd) and type:

```
python -m venv env
```

- **macOS**: Open Terminal and type:

```
python3 -m venv env
```

This command creates a directory named env (you can choose a different name if you like), which contains your virtual environment.

4.4 Activate the Virtual Environment

To start using the virtual environment, you need to activate it:

- **Windows**: Open Command Prompt (cmd) and type:

```
.\env\Scripts\activate
```

- **macOS**: Open Terminal and type:

```
source env/bin/activate
```

After activation, your terminal prompt should change, showing the name of the virtual environment. This indicates that the virtual environment is now active.

5. Installing Essential Dependencies Using pip

With your virtual environment activated, you can now install the necessary Python packages.

pip is the package installer for Python. You can install packages by typing:

```
pip install <package_name>
```

in terminal (please use specific package_name).

For example, to install torch, run:

```
pip install torch
```

You can use the pip install command to install the packages you'll need in the future.

Here's a list of Python packages commonly used in Machine Learning.

1. **numpy**: Fundamental package for numerical computing in Python.

```
pip install numpy
```

2. **pandas**: Essential for data manipulation and analysis, especially with tabular data.

```
pip install pandas
```

3. **torch (PyTorch)**: A deep learning framework widely used for building neural networks.

```
pip install torch
```

4. **torchviz**: A package for visualizing PyTorch model structures.

```
pip install torchviz
```

5. **scikit-learn**: A comprehensive library for classical machine learning algorithms.

```
pip install scikit-learn
```

6. **matplotlib**: A versatile plotting library for creating static, animated, and interactive visualizations.

```
pip install matplotlib
```

7. **seaborn**: Built on top of matplotlib, it provides a high-level interface for drawing attractive and informative statistical graphics.

```
pip install seaborn
```

Conclusion

Congratulations! You've successfully set up your first Python virtual environment and installed essential dependencies.