12 - env setup kaggle and colab usage tips

03 November 2024 02:58

Machine Learning Setup Using Anaconda & Jupyter Notebook

1. Introduction to ML Tools & Environment

- Machine learning (ML) involves numerous tools, among which **Anaconda** and **Jupyter Notebook** stand out for their popularity in data science.
- Anaconda simplifies installing and managing libraries for ML tasks, offering an integrated environment.

2. Setting Up Anaconda

- Download **Anaconda** from the official website.
- Install Anaconda following prompts (accepting terms, selecting installation paths).
- Anaconda Navigator opens, giving access to tools like Jupyter Notebook, Spyder, etc.

3. Tools in Anaconda

- **Jupyter Notebook**: Preferred for interactive coding, especially in data science.
- **Spyder**: An IDE within Anaconda; some may prefer it over Jupyter for scripting, though Jupyter offers extensive markdown and cell-based coding.

4. Using Jupyter Notebook

- Open Jupyter Notebook from Anaconda Navigator.
- Create a new folder, organize files and folders within it.
- Markdown for documentation and code cells for running scripts.
- Run commands with Shift + Enter; edit markdown for formatting.

5. Data Handling with Jupyter Notebook

- Import and explore datasets using pandas.
- Organize data processing into cells to allow step-by-step execution.
- Jupyter Notebook's Markdown helps annotate and document every step for reproducibility.

6. Creating Virtual Environments

- **Why Virtual Environments?** Keeps dependencies isolated for different projects, avoiding conflicts.
- Command: conda create --name <environment_name> python=<version>
- Activate environment: conda activate <environment_name>
- Deactivate environment when done.

7. Installing Libraries

- In the new environment, use conda install <package_name> to add required libraries, like NumPy, pandas, matplotlib, etc.
- Jupyter Notebook can also be installed specifically within the environment using conda install jupyter.

8. Importing and Working with Large Datasets

- **Example**: if we want to use the colab but dataset is in kaggle then use the following commnds in the colab

```
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

Then copy the data set api

```
!kaggle datasets download -d wobotintelligence/face-mask-detection-dataset
```

Then directly unzip

```
import zipfile
```

```
zip_ref = zipfile.ZipFile('face-mask-detection-dataset.zip', 'r')
zip_ref.extractall('/content')
zip_ref.close()
```

- **Unzipping large datasets**: Unzip commands to handle large datasets for efficient data management.

9. Google Colab as an Alternative

- **Benefits**: Uses Google's GPU, faster processing for deep learning.
- Directly connected to Google Drive; convenient for storing and accessing files.
- Google Colab offers similar functionality to Jupyter Notebook but with added cloud capabilities.

10. Practical Tips for Machine Learning Projects

- Use **virtual environments** for each project.
- Ensure **required dependencies** are specific to the environment, especially when deploying models.
- Document code effectively for **collaboration** and **future reference**.