

Setup & Installations

Prerequisites

To ensure a seamless learning experience and to make the most of this course, participants are expected to come with a foundational knowledge in the following areas:

1. **Programming Proficiency in Python:**
A strong command over Python's basic constructs, including functions, classes, and modules. Familiarity with libraries like NumPy, Pandas, Matplotlib is essential, as they form the backbone of many data manipulation tasks in AI.
2. **Deep Learning Framework - Tensorflow/PyTorch:**
A working knowledge of the TensorFlow (or PyTorch) framework is crucial, as many topics will delve into its functionalities and methods. Understanding TensorFlow's basic operations, data handling, and model building mechanisms will be invaluable.
3. **Basic Shell Commands:**
Comfortability in navigating the command-line interface (CLI), executing shell commands, and performing basic file operations are foundational for many MLOps tasks.
4. **Basic Data Structures:**
A good grasp of Python's primary data structures, especially dictionaries and lists, will be instrumental in understanding and manipulating data.
5. **File I/O:**
Knowledge of basic file input/output operations in Python, including reading from and writing to files, is vital for tasks involving data storage and manipulation.
6. **General AI and ML Concepts:**
While this course is centered around MLOps, a basic understanding of AI and machine learning concepts, including what models are and how they are trained, will set the context for many advanced topics.

It's important to note that while prior knowledge in these areas will provide a solid foundation, the course has been structured to ensure gradual progression. Even if you're not an expert in all of the prerequisites, a willingness to learn and engage actively in the course's hands-on components will be crucial for success. If you find yourself struggling with some concepts, we encourage leveraging the course resources, attending office hours, and participating in peer discussions to reinforce your understanding.

Software Installs

- IDE of choice or VS Code
- Docker Desktop
- Git or GitHub Desktop

Software Services

- Google Drive (Project documents, presentations, etc.)
- Communication Tools (WhatsApp, Slack, Messenger)
- GitHub account & repo for project source code
- Colab
- Google Cloud Platform account (Buckets & Deployment)

IDE

An integrated development environment (IDE) is a software platform for building software with various developer tools into a single graphical interface.

Examples:

- VS Code: <https://code.visualstudio.com/>
- PyCharm: <https://www.jetbrains.com/pycharm/>
- WebStorm: <https://www.jetbrains.com/webstorm/>
- Atom: <https://atom.io/>
- Sublime Text: <https://www.sublimetext.com/>

Installing VSCode:

Follow the instructions for your operating system from <https://code.visualstudio.com/download>

Notes for VS Code:

VS Code - Make sure to set the “LF” is selected in “Select End of Line Sequence”

Settings > Text Editor > Files > Eol > `\\n`

Docker

Docker is a platform for developing, shipping, and running applications in a containerized fashion. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.

Install links:

<https://docs.docker.com/docker-for-mac/install/>
<https://docs.docker.com/docker-for-windows/install/>
https://docs.docker.com/toolbox/toolbox_install_mac/
https://docs.docker.com/toolbox/toolbox_install_windows/

Linux:

<https://docs.docker.com/engine/installation/linux/ubuntu/>
<https://docs.docker.com/engine/installation/linux/debian/>
<https://docs.docker.com/engine/installation/linux/rhel/>
<https://docs.docker.com/engine/installation/linux/centos/>
<https://docs.docker.com/engine/installation/linux/fedora/>
<https://docs.docker.com/engine/installation/linux/linux-postinstall/>

Special notes for installing Docker on windows:

Install the docker desktop according to your windows edition. The windows home edition requires an additional install of W2LS, which will be taken care of by the installer itself. Make sure to have checked the option for W2LS installer in the start screen for docker desktop installer.

A restart for the machine will be prompted once docker desktop is installed (note : don't worry if W2LS is not installed yet). Once the machine restarts and you start docker desktop, a prompt for installing W2LS will be thrown if it isn't installed yet. Follow the instructions and get that installed as well. Make sure to have Docker Desktop running in the background, it can be checked by the presence of it's icon in the tray of icons on the taskbar.

*** If you are using VScode for editing files which will be executed inside the container, it is imperative that you have your default EoL set as LF instead of CRLF. This can be changed at the bottom right of the VScode editor window.

Other references for windows specific installation:

<https://docs.docker.com/docker-for-windows/wsl/>
<https://docs.microsoft.com/en-us/windows/wsl/tutorials/wsl-containers>
https://adamtheautomator.com/how-to-set-up-docker-in-wsl-step-by-step/#Using_the_WSL_Command-Line_Tool
<https://www.virtualizationhowto.com/2021/06/install-docker-on-windows-subsystem-for-linux-wsl-2-windows-10/>

Other SO posts:

<https://stackoverflow.com/questions/61920866/difference-between-wsl2docker-desktop-windows-and-directly-install-docker-on>

Simple Container:

Use a simple example docker container

- Run a container
- Execute shell command inside container

Docker image:

https://hub.docker.com/_/python?tab=description&page=1&ordering=last_updated

Open a command prompt and run this command:

```
docker run --rm --name test-ac215 -ti --entrypoint /bin/bash python:3.8-slim-buster
```

This will run a container and put you into the docker shell.

Execute the following commands inside the shell of the container:

```
ls
```

```
python --version
```

```
cat /etc/os-release
```

Google Drive

Collaborative environment for sharing project documents:

- Documents
- Presentations
- Colab notebooks

Communication Tools

- WhatsApp
- Slack
- Messenger

GitHub

GitHub is a hosted version control system and helps teams:

- Collaborate development efforts
- Keeps track of code versions
- Allows for both private/ Public repositories
- Track Changes in your code across versions
- Documentation
- Showcase Your Work
- Allows for Markdown, Notebooks
- Integration & automation with other tools

You will be required to create private github repositories.

Colab

- Colab is a Jupyter notebook environment provided by Google that runs entirely in the cloud.
- It does not require any setup and the notebooks that you create can be simultaneously edited just like in Google Docs.
- Colab comes pre-installed with many popular machine learning libraries which can be easily loaded in your notebook.
- Colab gives you access to free GPUs and TPUs for training

All our labs and homeworks in the deep learning module will use Colab.

https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index

Common instructions for Colab:

- Cloning a notebook
 - Select "File" menu and pick "Save a copy in Drive"
- Setting up GPU on a notebook
 - Go to the "Runtime" menu and select "Change runtime type". Then in the popup in "Hardware accelerator" select "GPU" and then click "Save"
 - If you want high RAM there is an option for that

Google Cloud Platform

- It is a suite of cloud computing services offered by Google.
- Consists of infrastructure as a service and platform as a service available as a pay per use option.
- Using GCP or any cloud platform is an advantage to scale up and down services very easily.

For this course you will need to create an account at: <https://console.cloud.google.com/>

We are working on getting GCP credits for you.

Also if you are interested and not already used them GCP has a Google Cloud Free Program where you can get free credits: <https://cloud.google.com/free/docs/gcp-free-tier>