

Build a Deep Learning Framework

Due Date - 3rd January 2021

OVERVIEW

In this project we will learn about the inner workings of famous Deep Learning Frameworks (PyTorch & TF). How they handle datasets, split data, help us design different architectures and finally train and test our models.

GOALS

1. Learn how a DL Framework is made and designed
2. Design our own Deep Learning framework consisting of the basic building blocks
3. Build a Neural Network and train it until we reach a good accuracy..

SPECIFICATIONS

- Simple libraries are the only one allowed (numpy, matplotlib, pandas and pillow)
- Every team should consist at max from 8 students.
- Meaningful commits and good software engineering (branching, release tags & project structure) will be an Evaluation criteria.
- A documentation for the framework modules will be required using GitHub Pages.
- The project will also required to be installable through pip install (Hint: use setuptools and setup.py for this)
- Presentations will be conducted with TAs for discussion and evaluation on day 4 and 5 in January 2021.

DELIVERABLES

- The main blocks that will need to be delivered is
 - A data module to read and process datasets
 - A NN module to design different architectures
 - Layers & Activation Functions
 - Losses & Evaluation Metrics

-
- An **optimization** module for training
 - A **visualization** module to track the training and testing processes
 - A **utils module for reading and saving models**
 - It will also be required to **build a Neural Network** with the framework and check all these points
 - Download and Split a dataset (**MNIST** or **CIFAR-10**) to training, validation and testing
 - **Construct an Architecture** (**LeNet** or **AlexNet**) and make sure all of its components are provided in your framework.
 - Train and test the model until a **good accuracy is** reached (**Evaluation Metrics** will need to be implemented in the framework also)
 - **Save the model into a compressed format**

The framework will be delivered in a GitHub Repo and the Neural Network will be delivered in a Notebook (online on colab or offline on Jupyter).

Competition

Stay Tuned...

References

1. LeNet Paper, [Gradient-Based Learning Applied to Document Recognition](#)
2. [LeNet5](#), [AlexNet](#) Architecture
3. [Deep Learning with PyTorch Book](#)
4. [Design Philosophy of Tensorflow](#) - Introduction to Tensorflow Part 1
5. [PyTorch](#), [Tensorflow](#) Docs
6. [MNIST](#), [CIFAR-10](#) Dataset
7. [Cuda in your Python](#)
8. [Evaluation Metrics for Classification](#)

-
9. [Save and Load Machine Learning Models](#)
 10. [Publish Your Project Documentation with GitHub Pages](#)
 11. [How to write your own Python Package and publish it on PyPi](#)
 12. [Structure your Python Project](#)
 13. [How to build a DL framework ?](#)
 14. [Live Graphs with Matplotlib](#)