Practical work 06 – 25/03/2025
Deep Learning Frameworks

Objectives

The main objective of this PW is to understand the overall functioning of the DL-frameworks PyTorch and Keras API for TensorFlow, as well as the visualisation tool TensorBoard. In particular we want to configure a multi-layer perceptron (MLP) and train it on the CIFAR10 image dataset using both frameworks.

Submission

— **Deadline** : Tuesday 1st April, 15pm

— **Format** :

  — Jupyter notebooks - one for each framework - with the necessary code to fulfil the exercise requirements. No notebook template is given, it is up to you to prepare the required information.

Submission of all files in a single zip-file using the naming convention (for team of two students #1, #2) :

```
family name_given name #1- family name_given name #2.zip
```

Exercise 1 MLP with PyTorch for CIFAR10

Based on the notebooks provided and discussed during the lecture, set up a notbook for the configuration and training of a MLP on the CIFAR10 images. These images (https://www.cs.toronto.edu/~kriz/cifar.html) of size 32x32 pixel show 10 different object categories. Figure 1 shows a mixure of all categories and Figure 2 the 'cat'-category only. A certain challenge represents the fact that these are 3-channel colour images.
The dataset is avaialble - as our (Fashion)MNIST data - in the `torchvision` library and can be downloaded from there (https://pytorch.org/vision/stable/generated/torchvision.datasets.CIFAR10.html#torchvision.datasets.CIFAR10). Feel free to use another source but it must be publicly available.

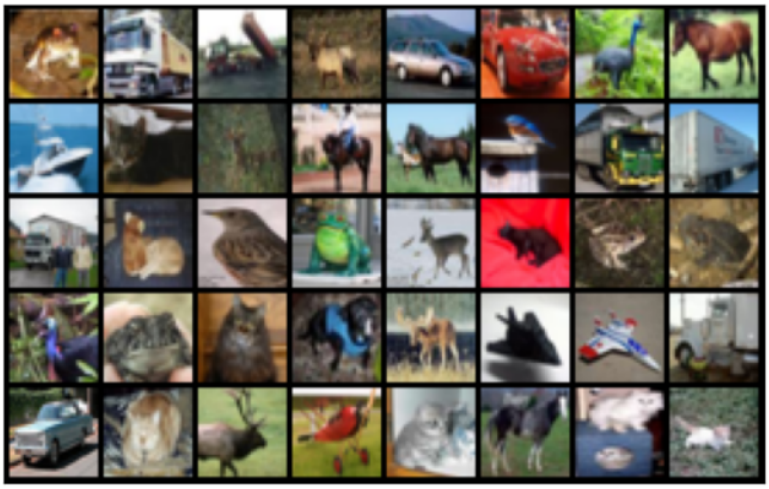

FIGURE 1 – Set of CIFAR10 images with mixture of all 10 categories.

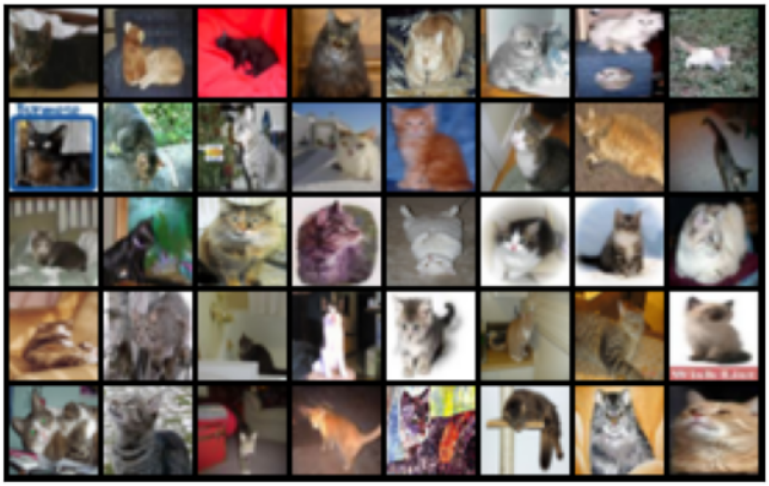

FIGURE 2 – Set of CIFAR10 images with 'cat'-category only.

Your implementation should provide the following features (guideline may be the notebooks `6.4.mlp_pytorch_3_stud.ipynb` or `6.5.mlp_pytorch_final.ipynb`) :

- Download of the CIFAR10 images form an appropriate server and local storage for further usage.
- Configuration of a custom `Dataset` (c.f. chapter 6.2.2 in the lecture notes) that will provide the CIFAR10 data to the DataLoader class during training.
- Set up of a `SummaryWriter` for TensorBoard (c.f. chapter 6.2.3 in the lecture notes) and output of a set of CIFAR10 images (as grid like the figures above) to TensorBoard.
- Set up of a MLP with confiigurable number of hidden layers that takes CIFAR10 (colour) images as input.
- Preparation of the `SummaryWriter` for continuous output of training and validation loss/error to TensorBoard during the training process.

Perform a few training runs to validate your implementation. Don't expect the accuracy to be very high, it will be around 50% for all 10 classes.

Exercise 2 MLP with Keras API for TensorFlow (optional !)

Repeat the exercise above now using the Keras API for TensorFlow (guideline may be the notebook `6.6.mlp_tensorflow_stud.ipynb`). The implementation will be somewhat simpler because you can use direclty the full dataset (without split in train and validation set) as input (as numpy array) and becauseTensorBoard is alreday integrated into TensorFlow.
Your implementation should provide the following features :

- Download of the CIFAR10 images form an appropriate server and local storage for further usage.
- Set up of a MLP with configurable number of hidden layers that takes CIFAR10 (colour) images as input.
- Preparation of TensorBoard for continuous output of training and validation loss/error to TensorBoard during the training process.

Again perform a few training runs to validate your implementation.