Home assignment - Machine Learning (ITLB356, MIB)

Task description

Challenge: Get the best results on the rock paper scissors dataset!

> *"Rock Paper Scissors contains images from various hands, from different races, ages, and genders, posed into Rock / Paper or Scissors and labeled as such. You can download the training set here and the test set here. I created these images using CGI techniques as an experiment in determining if a model trained on a CGI-based dataset could classify real images. I also generated a few pictures that you can use for predictions. You can find them here. Note that all of these pictures use a plain white background. Each image is 300×300 pixels in 24-bit color."*

Your task is as follows:

- Download the dataset from tensorflow_datasets like:

```
import tensorflow_datasets as tdfs

DATASET_NAME = 'rock_paper_scissors'

(dataset_train_raw, dataset_test_raw), dataset_info = tfds.load(

    name=DATASET_NAME,

    data_dir='tmp',

    with_info=True,

    as_supervised=True,

    split=[tfds.Split.TRAIN, tfds.Split.TEST],

)
```

- Convert from TensorFlow dataset to NumPy array if you want:

```
def dataset_to_numpy(ds):
    """
    Convert TensorFlow dataset to NumPy arrays
    """
    images = []
    labels = []
    # Iterate over a dataset
    for i, (image, label) in enumerate(tfds.as_numpy(ds)):
        images.append(image)
        labels.append(label)
    for i, img in enumerate(images):
        if i < 3:
            print(img.shape, labels[i])

    return images, labels
```

- Split to training, validation, and test sets.
- Challenge: Get the best results on the rock_paper_scissors dataset!

- The challenge of this task is to get the best results on the rock_paper_scissors dataset by tuning hyperparameters of a NN model and observing convergence behavior.
- Best - for simplicity - means the highest accuracy on the validation set.

Added constraint: The model with the "best performance" has to be saved, so it should not be just a printout happening once during training!

You may NOT manipulate the validation set!

Please observe the following:

- You must use a single standalone Jupyter Notebook to solve the task and submit
  - the .ipynb file, and
  - a .pdf file generated from the notebook.

Note for those working on Google Colab: a link to your notebook will not suffice: you have to download and submit the file itself.

Follow the principle of literate programming, and make use of the markdown cells of the notebook.

Deadline

Please refer to the Moodle page of the module.

Assessment

The assignment will be assessed based on the following criteria (see the grid on Moodle):

- Specification fulfillment (50%)
- Literate programming and markdown cells (20%)
- Conceptual understanding (20%)
- Clean code (10%)

The **resit arrangement** for the assignment is the same as above; you may resubmit the same paper, with corrections, that you submitted by the original deadline. The resubmission deadline will be specified on Moodle after the grades for the original submission are published.

Upload all your files to Moodle.

Academic conduct notice

Where the Academic Conduct Officer has reason to suspect that a piece of work submitted by a student was wholly or in part written by someone other than the student who submitted it, and this has not been disclosed by the student, they may call for the student to defend the work in **viva or a written comprehension test**. The burden of proof in such a viva or test will be upon the student to demonstrate to the examination panel's satisfaction his/her full comprehension of the work s/he has submitted. Failure to appear without satisfactory explanation will result in immediate failure of that assessment, with consequences of academic misconduct and application of sanctions.