**Home assignment - Deep Learning (ITLB359, MIB)**

* 1. **Task description**

Challenge: Get the best results on the [malaria](https://www.tensorflow.org/datasets/catalog/malaria) dataset!

"*Malaria contains images from various cells. Some cells are infected with malaria, other cells are not. You can download the dataset. Each image is a different shape in 24-bit colour."*

Your task is as follows:

* Download the dataset from google drive like:

| !wget "https://drive.google.com/uc?id=1dl6KXbpRrUNrH5xC26MpYq3rHjOewUTF&export=download&authuser=0" -O malaria.zip  !unzip malaria.zip  %cd malaria |
| --- |

* Convert from image dataset to NumPy array for training:

| list\_of\_class1 = os.listdir("Parasitized")  list\_of\_class0 = os.listdir("Uninfected")  labels1 = np.ones(len(list\_of\_class1))  labels0 = np.zeros(len(list\_of\_class0))  labels = np.concatenate((labels1, labels0), axis=0)  dataset = []  dim = (100, 100)  for img in list\_of\_class1:  image = cv2.imread("Parasitized/" + str(img))  resized = cv2.resize(image, dim, interpolation = cv2.INTER\_AREA)  dataset.append(resized)  for img in list\_of\_class0:  image = cv2.imread("Uninfected/" + str(img))  resized = cv2.resize(image, dim, interpolation = cv2.INTER\_AREA)  dataset.append(resized) |
| --- |

* Split to training, validation and test sets.
* Challenge: Get the best results on the malaria dataset!
* The challenge of this task is to get the best results on the malaria dataset by tuning hyperparameters of a NN model and observing convergence behavior.
* Best - for simplicity - means the highest accuracy on the validation set.
* Reflection
  + Critically evaluate your work, including other approaches.
  + Answer in a markdown cell.

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 (that is, you cannot pick and choose which samples belong to the validation set to make your model “perform” better on validation data)!**

**Please observe the following:**

* You must use a single standalone Jupyter Notebook to solve the task and submit the .ipynb file.

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

1. 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.

* 1. **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 grounding (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 your file (.ipynb) to Moodle.

* 1. **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.