University of St. Gallen
Institute of Computer Science

Fundamentals of Computer Science (3,125), HS2019
H. Hemati, K. Schürholt, D. Borth
konstantin.schuerholt@unisg.ch

Assignment 11: Machine Learning III - Neural Networks

Deadline: Dec 31, 2019; 12:00 CET

Introduction In the last two weeks, you have learned to evaluate the performance of classifications and built your own classifiers. In this last exercise, we will train neural networks to classify images. We will go through the entire workflow using the pytorch library.

Rationale Neural networks are the state-of-the-art in classification. While the underlying mechanisms are quite involved, libraries like pytorch or tensorflow provide powerful tools that make machine learning accessible to the broader public. These libraries can be used for both research as well as production; they offer wrappers with convenient easy-access solutions, but also make every detail available for experienced developers. It is therefore very valuable to learn to develop simple models in pytorch.

Prerequisites If you decide to run your code on your local machine, install pytorch by executing: `conda install pytorch torchvision -c pytorch`. Use your favorite method to open a Jupyter Notebook, e.g. any of these options below:

- local: run **jupyter lab** in your terminal.
- local: open the *.ipynb file in visual studio code.
- cloud: open https://notebooks.azure.com, login with your university account, create and run a new project and upload the *.ipynb file. To avoid compatibility issues, change the kernel to python 3.6 (Kernel: change kernel: python 3.6).
- cloud: use any other jupyter notebook environment.

This week, the notebook contains not only code and instruction, but also background information and motivation for the methods we introduce.
As before, for this assignment, you **ONLY** need to edit the notebook!
To run your code, click into one of the cells (the blocks of text that you can edit) and then click the *Run* button on the side of the cell.

(1) Neural Network Training (8 Points)

In the programming task, you will create, train and evaluate a neural network.

1.) implement the `load_data` method.

2.) complete the `__init__` and `forward` methods of the `Model` class.

3.) complete the train and test part of the `train_and_test` method.

4.) implement the `plot_performance` method.

(2) Neural Networks - Understanding (2 Points)

1.) read and understand the content we provide throughout the notebook.

2.) answer the multiple choice questions in the jupyter notebook.

Hand-in Instructions Create a file `README.txt` that includes 1.) any reference that you used to complete this assignment, 2.) pitfalls you encountered and 3.) short explanations of your solution if necessary. Fill in your name and student ID in the comment header of files you edited.

Depending on your environment, export or download your *ipynb **and** *py files, which contains the same code. Include both of these files in your `zip` file. Also, by running the cell with the `train_and_test` function, you also save a copy of your entire model in `model_checkpoint.pt`. Download that model and add it to the zip file. In your last task, you create two figures with the training loss and test accuracy. Kindly also include these figures in your zip.

Compress the whole folder with the Python files and the `README.txt` to a `zip` file named "`assignment11.zip`". Upload the zip file to your exercise group's Course page on Canvas. See Figure.1 for the list of files that should be included in the zip file.

```
assignment11.zip
├── README.txt
├── assignment_11.ipynb
├── assignment_11.py
├── model_checkpoint.pt
├── loss.png
└── accuracy.png
```

Figure 1: The required files for submission.