

General constraints for code submissions Please adhere to these rules to make our and your life easier! We will deduct points if your solution does not fulfill the following:

- If not stated otherwise, we will use exclusively Python 3.6.
- If not stated otherwise, we expect a Python script, which we will invoke exactly as stated on the exercise sheet.
- Your solution exactly returns the required output (neither less nor more) – you can implement a `--verbose` option to increase the verbosity level for developing.
- Add comments and docstrings, so we can understand your solution.
- (If applicable) The **README** describes how to install requirements or provides addition information.
- (If applicable) Add required additional packages to **requirements.txt**. Explain in your **README** what this package does, why you use that package and provide a link to it's documentation or GitHub page.
- (If applicable) All prepared unittests have to pass.
- (If applicable) You can (and sometimes have to) reuse code from previous exercises.

As you learned in the last lecture we can use Grey-Box Optimization to make the hyperparameter search more efficient. For this we assume that we have some information directly from the optimization process. These could be for example:

- Early stopping poorly performing configurations
- Cheap-to-evaluate proxies
- Gradients

In this exercise you will implement a straight-forward early stopping algorithm commonly used in practice called Successive Halving and its popular extension Hyperband.

1. Grey-Box Optimization for HPO

[14 points]

We provide you with a rough structure of Successive Halving and Hyperband. You will implement the remaining parts to **minimize** an objective function.

- Implement Successive Halving based on the algorithm you have seen in the lecture. Your implementation should satisfy the test in `test_successive_halving.py`. [5pt.]
- Please add a plot of the Successive Halving results using the utility function provided to the repo. [2pt.]
- Implement Hyperband by reusing the Successive Halving implementation based on the algorithm discussed in the lecture. Your implementation should satisfy the test in `test_hyperband.py`. [4pt.]
- Please add a plot of the Hyperband results using the utility function provided to the repo. [2pt.]
- What are the advantages/disadvantages of Successive-Halving/Hyperband compared to Bayesian Optimization? [1pt.]