Prof. Dr. Holger Hoos
Chair for AI Methodology
RWTH Aachen University

Hyperparameter Optimisation for Machine Learning Assignment 1

Hadar Shavit, Nick Kocher and Holger Hoos

Introduction

The goal of this assignment is to understand four different hyperparameter optimisation techniques: random search, grid search, successive halving and Bayesian optimisation. You will apply these techniques to multiple machine learning algorithms and compare their performance.
Deadline: 21 May 2025, 23:59 CEST

Deliverables

- A 2-page report includes a short introduction, description of the methods implemented (1-2 sentences per method), experimental setup, results (with description of the results) and conclusions. The 2-page report should contain all the figures for the mandatory parts of the assignment. If you perform additional experiments, you can put the figures and description in the appendix; however, the 2-page report must be self-contained. The report should use the LaTeXtemplate from Moodle. You are not allowed to change the font size or margins but can resize the figures to be small (but not too small).
- Code for the methods implemented, including comments for each method (what are the inputs, outputs what is it doing) and for the non-trivial code lines.

Methods to implement

In the zip file accompanying the assignment, you can find four files for the methods you will implement in the assignment: RANDOM_SEARCH.PY, GRID_SERACH.PY, GRID_SERACH.PY, SUCCESSIVE_HALVING.PY. In each file there is a class with several empty methods:

- __INIT__: where you should initialise the method
- ASK: return the next configuration to sample
- TELL: get the value of a configuration

The first part of the assignment is to implement random search, grid search and Bayesian optimisation within the provided framework. You are allowed to add additional methods to the classes. However, the three methods described above must be implemented to perform the desired functionality. For Bayesian optimisation, you should use GAUSSIANPROCESSREGRESSOR from SCIKIT-LEARN as a surrogate model. You are not allowed to use any pre-made code for the implementation of the acquisition function or the acquisition function optimisation. You can decide on the other components of the algorithm, but make sure to report and justify the selection in your report.
When implementing the methods, you can use the ConfigSpace library; however, you are not allowed to use methods such as SAMPLE_CONFIGURATION, GENERATE_GRID or GET_ARRAY. You are

not allowed to use any other packages other than the ones in the requirements.txt file. If you have questions, please get in touch with us.

Experiments

We provide an additional EXPERIMENTS.PY file with a while loop that calls the methods you implemented and saves the results. The script is unfinished. You should decide on an experimental setup, including the evaluation budget that correctly compares the four methods, and a way to deal with the randomness of the algorithms. Your task is to modify the file to benchmark the methods on two scenarios from YAHPO [1]: CIFAR10 from NB301 and dataset 16 of rbv2_xgboost. You are allowed to add additional arguments, loops and other modifications to this file.
Once you are done running the experiments, you are free to create the figures however you wish. However, remember to adhere to the visualisation techniques we showed during the first meeting!

Provided files

We provide the following files:

- HPO_ALGORITHM.PY
- BAYESIAN_OPTIMISATION.PY
- GRID_SEARCH.PY
- RANDOM_SEARCH.PY
- SUCCESSIVE_HALVING.PY
- EXPERIMENT.PY

You are allowed to add additional files, methods, class attributes etc.. However, for each HPO method you implement, the methods __INIT__, ASK and TELL should follow the API as defined in HPO_ALGORITHM.PY. You are not allowed to modify this API. In case you require additional parameters for one of the methods you implement (for example, the η parameter of successive halving), you are encouraged to add it as a parameter to the __INIT__ method with a default value.

References

[1] Florian Pfisterer et al. “YAHPO Gym - An Efficient Multi-Objective Multi-Fidelity Benchmark for Hyperparameter Optimization”. In: *Proceedings of the International Conference on Automated Machine Learning (AutoML)*. Vol. 188. PMLR, 2022, pp. 3/1–39.