

# Compilers meet Machine Learning

Bachelor/Master/Hiwi-Jobs available in a joint project by  
Jun.-Prof. Sebastian Hack and Jun.-Prof. Matthias Hein

Code optimization is a hard task. Even under very rough optimality criteria, many optimizations are NP-complete or even uncomputable. Therefore, modern compilers apply a sequence of heuristics to accelerate the program's runtime. Currently, these heuristics are tuned *manually* by an expert, usually on a very limited set of benchmarks. This has several disadvantages:

- The relation between program code and the use of certain optimizations is quite complex. Therefore, hand-tuned heuristics built on a set of simple decision rules are usually suboptimal or even detrimental, in particular since there is no systematic tuning of the parameters.
- Even if the hand-crafted rules perform well on a small set of benchmark programs, this does not imply that they will work well on "all possible code". In other words: the statistics of code features of benchmark programs is not representative for the set of code which will be compiled in practice.
- Whenever support for a new target is integrated into the compiler, the tuning has to be redone from scratch. This is a very time-consuming and thus costly procedure.

The goal of this project is to overcome these problems using machine-learning techniques to put the tuning of compiler heuristics on a statistically sound foundation.

## Required skills:

- interest to explore a new emerging field of research
- you enjoy writing fast code and getting things to work
- good programming skills in C/C++ and some scripting language
- basic knowledge / interest how compilers work

## Useful skills:

- familiarity with performance tools like valgrind
- background in machine learning

**Contact:** By email to [hack@cs.uni-sb.de](mailto:hack@cs.uni-sb.de) and [hein@cs.uni-sb.de](mailto:hein@cs.uni-sb.de)