MASTER'S THESIS

SOFTWARE 2.0: PRAGMATIC DIFFERENTIAL PROGRAMMING IN MODEL-DRIVEN SOFTWARE ENGINEERING

Contact

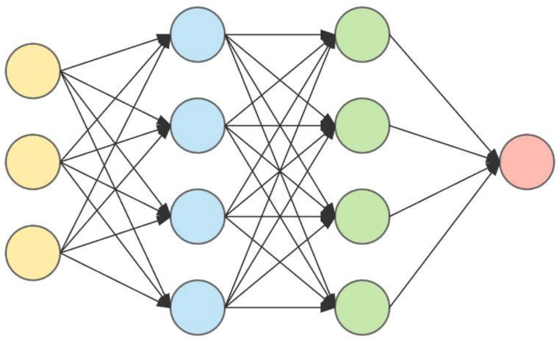
Dipl.-Ing.
Evgeny Kusmenko
Tel.: 0241 80 21342
kusmenko@se-rwth.de

Dr. rer. nat.
Andreas Wortmann
Tel.: 0241 80 21346
wortmann@se-rwth.de

Statement of Task

Prof. Dr.
Bernhard Rumpe
Tel.: 0241 80 21301
rumpe@se-rwth.de

Machine Learning enables us to train algorithms based on examples in various application domains. This is accomplished by designing parameterized models and adapting their parameters using a training data set. The resulting trained models can be integrated into larger software architectures as building blocks to deal with specific problems, e.g., image detection, automated image recognition, decision making, and more. At the Chair of Software Engineering we want to extend this paradigm by making all parts of a software trainable. This can be accomplished using concepts of differential programming by making all constructs of a software language, such as conditionals, loops, etc. differentiable.



The goal of this thesis is to develop and implement concepts for high-level differentiable constructs and to integrate them into two host-languages. In a bottom up approach, the student is going to extend the graph-oriented deep learning language MontiAnna by trainable conditions, loops, and the like to facilitate the graph construction of a deep learning model. Furthermore, a top-down approach will be used to enrich an imperative language by similar differential constructs to render the definition of a highly adaptive software architecture possible. Afterwards, the resulting languages will be evaluated and compared against each other with regard to a set of software language evaluation criteria in the scope of a larger case study.

The proposed thesis is highly research-oriented. We expect the candidate to be able to work highly autonomously, to employ new technologies and frameworks, and to develop a profound understanding of the underlying theory.

NECESSARY PRIOR KNOWLEDGE

- Java, C++, Python
- Machine Learning basics, particularly Deep Learning

BENEFICIAL PRIOR KNOWLEDGE

- At least one deep learning framework, e.g., Tensorflow or MxNet