Fourth MODE Workshop on Differentiable Programming for Experiment Design

Contribution ID: **36**

Type: **not specified**

Advanced optimizations for source transformation based automatic differentiation

Wednesday 25 September 2024 11:50 (20 minutes)

Advanced optimizations for source transformation based automatic differentiation

Clad is a LLVM/Clang plugin designed to provide automatic differentiation (AD) for C++ mathematical functions. It generates code for computing derivatives modifying abstract syntax tree using LLVM compiler features. Clad supports forward- and reverse-mode differentiation that are effectively used to integrate all kinds of functions. The typical AD approach in Machine Learning tools records and flattens the compute graph at runtime, whereas Clad can perform more advanced optimizations at compile time using a rich program representation provided by the Clang AST. These optimizations investigate which parts of the computation graph are relevant to the AD rules. One such technique is the "To-Be-Recorded"optimization, which reduces the memory pressure to the clad tape data structure in the adjoint mode. Another optimization technique is activity analysis, which discards all derivative statements that are not relevant to the generated code. In the talk we will explain compiler-level optimizations specific to AD, and will show some specific examples of how these analyses have impacted clad applications.

Author: ANDRIICHUK, Maksym

Co-authors: ZARYTSKYI, Petro; VASSILEV, Vassil

Presenter: ANDRIICHUK, Maksym

Session Classification: Computer Science

Track Classification: Computer Science