Presentation Abstract

Compiling ONNX Neural Network Model Using MLIR

Tung D. Le[1,a)] Gheorghe-Teodor Bercea[2] Tong Chen[2] Alexandre E. Eichenberger[2]
Haruki Imai[1] Tian Jin[2] Kiyokuni Kawachiya[1] Yasushi Negishi[1] Kevin O'Brien[2]

Presented: July 31, 2020

Neural network model is becoming popular and has been used in various tasks such as computer vision, speech recognition, and natural language processing. It is often the case that the training phase of a model is done in an environment, while the inference phase is executed in another environment. It is because the optimization characteristics for each phase are largely different. As a result, it is critical to efficiently compile a trained model for inferencing on different environments. To represent neural network models, users often use ONNX which is an open standard format for machine learning interoperability. We are developing a framework for compiling a model in ONNX into a standalone binary that is executable on different target hardwares such as x86, P, and Z. The framework is written using MLIR, a modern compiler infrastructure for multi-level intermediate representations. In particular, we introduce two internal representations: ONNX IR for representing ONNX operators, and Kernel IR as an intermediate representation for efficiently lowering ONNX operators into LLVM bitcode. In this presentation, we will discuss the overall structure of the framework and show some practical examples of converting ONNX operators and models. We also cover several issues related to endianness.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

[1] IBM Research - Tokyo, Chuo, Tokyo 103–8510, Japan
[2] IBM T.J. Watson Research Center, Yorktown Heights, NY 10598
[a)] tung@jp.ibm.com