Contribution ID: **214** Type: **Oral**

PHASM: A toolkit for creating AI surrogate models within legacy codebases

Thursday 27 October 2022 14:30 (20 minutes)

PHASM is a software toolkit, currently under development, for creating AI-based surrogate models of scientific code. AI-based surrogate models are widely used for creating fast and inverse simulations. The project anticipates an additional, future use case: adapting legacy code to modern hardware. Data centers are investing in heterogeneous hardware such as GPUs and FPGAs; meanwhile, many important codebases are unable to take advantage of this hardware's superior parallellism without undergoing a costly rewrite. An alternative is to train a neural net surrogate model to mimic the computationally intensive functions in the code, and deploy the surrogate on the exotic hardware instead. PHASM addresses three specific challenges: (1) systematically discovering which functions can be effectively replaced with a surrogate, (2) automatically identifying, for a given function, the true space of inputs and outputs including those not apparent from the type signature, and (3) integrating a machine learning model into a legacy codebase cleanly and with a high level of abstraction. In the first year of development, a proof of concept has been developed for each challenge. A surrogate API makes it easy to bring PyTorch models into the C++ ecosystem and uses profunctor optics to establish a two-way data binding between C++ datatypes and tensors. A model variable discovery tool performs a dynamic binary analysis using Intel PIN in order to identify a target function's model variable space, including types, shapes, and ranges, and generate the optics code necessary to bind the model to the function. Future work may include exploring the limits of surrogate models for functions of increasing size and complexity, and adaptively generating synthetic training data based on uncertainty estimates.

Significance

This is the first time presenting this project outside of Jefferson Lab. It already explores several novel technical approaches, specifically (1) using profunctor optics to create a two-way binding between arbitrary C++ datatypes and tensors, and (2) using dynamic binary analysis to discover the space of model variables within an unknown codebase.

References

No publications yet!

Experiment context, if any

Not used by any experiments yet!

Primary author: BREI, Nathan (Jefferson Lab)

Co-authors: MEI, Xinxin (Jefferson Lab); LAWRENCE, David (Jefferson Lab)

Presenter: BREI, Nathan (Jefferson Lab)

Session Classification: Track 1: Computing Technology for Physics Research

Track Classification: Track 1: Computing Technology for Physics Research