Extend the Automatic Differentiation Support in RooFit

Abhigyan Acherjee, University of Cincinnati

Mentors:
Vassil Vassilev, Research Software Engineer, Dept. of Physics, Princeton University
David Lange, Computational Physicist, Dept. of Physics, Princeton University
Jonas Rembser, Senior Applied Fellow, CERN

Background

Automatic differentiation is a set of techniques to compute analytic derivatives in a computer program. This is usually faster than numerical differentiation, and offers more scalability for problems with large numbers of parameters. It's also more accurate than numerical differentiation.

RooFit library is a toolkit for modeling the expected distribution of events in a physics analysis.The core functionality of RooFit is to enable the modeling of 'event data' distributions, where each event is a discrete occurrence in time, and has one or more measured observables associated with it – it uses a granular structure in its mapping of mathematical data models' components to C++ objects.

Problem Description

In terms of minimization time, Roofit offers faster results even with numerical differentiation techniques as compared to minimizing a likelihood function that is written by hand in C++, due its complex caching logic. Automatic differentiation gives an additional speedup and more accuracy and scalability for problems with large number of parameters. The purpose of this project will be to firstly use Minuit as an optimization algorithm with externally provided gradients, extend support to cover HistFactory and other parts of RooFit, and finally to optimize Clad generated derivatives and further explore how they can be parallelized (OpenMP or CUDA).

There have been various ongoing efforts into modernizing RooFit. For example -Cramer et al., where fitting work is distributed across computational nodes. In addition, there have been efforts to utilize better CPU vectorization and utilization of GPUs9 (Michalainas et al, 2023) Recently RooFit has enabled automatic differentiation with Clad (Singh et al., 2023). In essence, the complex object-oriented structure of the models in RooFit is transpiled into simpler C++. That is, the computation graph is traversed and "squashed" into a single routine. The new architecture enables application

of efficient AD algorithms and the results have shown faster and more accurate execution.“flattened”

However, the promising results need to be proved further on production-grade workflows. That is, implement enough RooFit primitives to support LHC-experiment scale analysis such as the ones that are based on HistFactory – a facility heavily used by the LHC experiments.

The foreseen work is in two major directions. First, extend the AD support in RooFit by transpiling more RooFit primitives. And second, profile and extend if necessary the underlying minimization infrastructure in the minuit package.

Goals

- Develop an understanding of the current codebase and experiment offering speedups through Automatic Differentiation.
- Understand the HistFactory models used in RooFit.
- Investigate the speedups offered through Minuit and Minuit optimisation strategies.
- Develop functionalities and classes to extend support to HistFactory and other parts of RooFit
- Profile the benchmarks and understand performance bottlenecks of the current AD-aware tutorials.
- Extend the Higgs to a more complex example for the Higgs to gamma-gamma analysis in rootbench
- Incorporate the AGC benchmarks in the public benchmark infrastructure.
- Support a tutorial that contains numeric integrals by using Clad's fallback mode to numerically differentiate only such integrals and use AD for the rest of the model.
- Investigate improving the Minuit seeding procedure.
- Explore opportunities for parallelism using OpenMP or CUDA.

Deliverables

Week 1 (14th-20th Aug)	Develop an understanding of the current codebase and experiment offering speedups through Automatic Differentiation. **Deliverable:** set up development environment.
Week 2 (21st-28th Aug)	Examine the AD-aware tutorials of RooFit. Enumerate a list of tutorials which need to support AD. **Deliverable:** a plan of tutorials to be supported and a detailed plan of primitives that need to be transpiled.

Week 3 (28th Aug-3rd Sept)	Investigate the status of AD support in the HistFactory benchmarking models. **Deliverable**: A list of improvements in the testing and benchmarking infrastructure to capture the AD case
Week 4(4th Sept -10th Sept)	Improve the benchmark infrastructure with focus on the AD use case by adding more tests. Deliverable: Add more tests in rootbench.
Week 5(11th Sept-17th Sept)	A Buffer week in case a task takes longer than anticipated
Week 6,7 () (18th Sept - 1st Oct)	Profile the benchmarks and understand performance bottlenecks of the current AD-aware tutorials. **Deliverable**: A report or presentation about where we spend time and a short report how to improve it
Week 8(2nd Oct - 8th Oct)	Develop functionalities and classes to extend support to HistFactory and other parts of RooFit. For example, support Higgs to gamma-gamma analysis (via the opendata website). **Deliverable**: RooFit support of that analysis demonstrated via unittest
Week 9 (9th Oct-15th Oct)	Extend the Higgs to a more complex example for the Higgs to gamma-gamma analysis in rootbench. **Deliverable**: Support all of the unsupported examples in https://github.com/root-project/rootbench/blob/master/root/roofit/atlas-benchmarks/download_workspaces.sh Namely, *WS-Comb-STXS_toy.root*, *WS-HGam-STXS_xs_toy.root*
Week 10 (16th Oct-22nd Oct)	Buffer
Week 11,12(23rd Oct-5th Nov)	Implement the fit from the IRIS HEP analysis grand challenge (AGC) in RooFit with the AD backend, implement missing support the needed RooFit primitives for that usecase. Benchmark and compare with other AGC implementations. **Deliverable**: technical report on the benchmark results
Week 13,14 (6th Nov-19th Nov)	Incorporate the AGC benchmarks in the public benchmark infrastructure. **Deliverable**: Benchmarks comparing RooFit AD with other fitting tools.
Week 15,16 (20th Nov-3rd Dec)	Support a tutorial that contains numeric integrals by using Clad's fallback mode to numerically differentiate only such integrals and use AD for the rest of the model.
3rd Dec-7th	(No activity due to winter break and finals week)

Jan	
Week 17,18 (8th Jan-21st Jan)	Investigate improving the Minuit seeding procedure. **Deliverable**: initialize the minimizer without having to numerically calculate the second derivatives for all parameters"
Week 1 (22nd Jan-28th Jan)	Buffer
Week 20 (29th Jan-5th Feb)	Prepare documentation and tutorials to provide support for the classes in the code squashing for AD.'
Week 21, 22 (5th Feb-18th Feb)	Explore opportunities for parallelism using Open MP/CUDA.
Week 23 (19th Feb-25th Feb)	Develop final documentation and tutorials. Deliverable: a document on how to write new RooFit primitive support; Technical document on the usage of AD in RooFit at scale.
Week 24 (26th Feb-4th March)	Prepare a final presentation.

References:

1. Michalainas, E. et a. (2023). *Acceleration with GPUs and other RooFit news*. IOP Science. https://iopscience.iop.org/article/10.1088/1742-6596/2438/1/012066
2. Singh, G., Rembser, J., Moneta, L., & Lange, D. (2023). *Automatic differentiation of binned likelihoods with Roofit and clad*, CHEP 2023 https://www.researchgate.net/publication/369855627_Automatic_Differentiation_of_Binned_Likelihoods_With_Roofit_and_Clad
3. Cramer, K., Verkerke, W., Moneta, L., & Shibata, A. (n.d.). *HistFactory: A tool for creating statistical models for use with ... - CERN*. http://cds.cern.ch/record/1456844/files/CERN-OPEN-2012-016.pdf?subformat=pdfa&version=1. https://cds.cern.ch/record/1456844/files/CERN-OPEN-2012-016.pdf?version=1