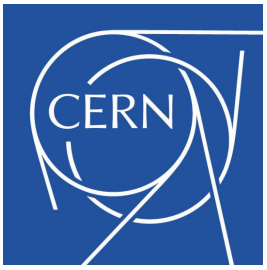
Google Summer of Code

2023

@



Proposal :

Add support for differentiating with respect to multidimensional arrays (or pointers) in Clad

By

Rishabh Bali (VJTI, University Of Mumbai)

Project Mentors : Dr Vassil Vassilev(vvasilev@cern.ch),
Parth Arora(partharora99160808@gmail.com

Abstract :

Clad is a clang plugin that enables automatic differentiation of mathematical functions for C++. Based on the LLVM compiler infrastructure clad does this by parsing and transforming the abstract syntax tree (AST) it detects the mathematical function and consequently inserts the derived function into the AST. Clad currently supports both forward and reverse mode automatic Differentiation along with other features like differentiating functors, differentiating w.r.t one dimensional arrays and it even has its own error estimation framework.

Even though clad is fast and efficient, the reverse and the forward mode lack support for differentiating w.r.t to multidimensional array; the support for differentiating w.r.t to pointers is limited as well.

This proposal aims to add support for differentiating multi dimensional arrays/pointers in clad.

2 . Deliverables

- Add support for differentiating w.r.t multidimensional arrays in forward mode of clad
- Add support for differentiation w.r.t pointers in the reverse mode of clad.
- Add support for differentiation w.r.t multi-dimensional arrays in the reverse mode in clad.

3. Project Details

3.1 Add support for differentiating w.r.t to multi-dimensional arrays using clad::differentiate.

Currently clad::differentiate has support for single dimensional arrays. An example is as follows :

```
float sum (float *y) {
  return  y[0] + y[1];
```

```
}

int main() {
   float y[2] = {1,2};
   auto d_fn = clad::differentiate(sum, "y[0]");
}
```

On attempting to differentiate w.r.t multi-dimensional arrays we get an error.

This error is due to the certain condition in ***"ForwardModeVisitor::Derive"***
We need to make additional changes in this method to enable differentiation w.r.t multi-dimensional arrays.

After adding support for multi-dimensional arrays in forward mode the following code should run without any errors.

```
float sum (float arr[2][2], float a) {
return arr[0][1] + arr[1][0];
}

int main(){
   float arr[2][2] = {{1., 2.}, {2., 4.}};
   auto d_fn = clad::differentiate(sum,"arr[0][1]");
}
```

3.2 Add support for differentiating w.r.t pointers in reverse mode AD.

Currently Reverse Mode AD doesn't support differentiation w.r.t pointers. We need to convert pointers to references before passing them as a parameter to the function. We need to extend the ***"ReverseModeVisitor"*** class to add support for differentiating w.r.t pointers.

3.3 Add support for differentiation w.r.t to multidimensional arrays in reverse mode.

Currently reverse mode AD doesn't support differentiation w.r.t multi-dimensional arrays.

```
double fn(double arr[2][2]) {
  double res = arr[0][0] + 2 * arr[1][1];
  return res;
}

int main() {
  auto d_fn = clad::gradient(fn);
  double arr[2][2] = {{2, 3}, {4, 5}};
  double d_arr[2][2] = {};
  d_fn.execute(arr, d_arr);
}
```

The following code snippet returns an error. This can be solved by adding support for multidimensional arrays to the reverse mode in clad. This can be achieved by extending the ***"ReverseModeVisitor"*** class to add support for multi-dimensional arrays.

Proposed Timeline :

The Timeline is flexible and can be adjusted accordingly if more things come up.
I have also left a few days before the evaluations free. This provides buffer-time and will ensure the timeline is being followed.

Week	Task to be completed
April 4 - May 4	Explore the codebase in more detail. Solve some more issues and add missing documentation.
Community Bonding Period Begins	Continue exploring the codebase and get in contact with the mentors to discuss the existing codebase and identify the basic blocks we need to add to the codebase to support differentiation w.r.t multi-dimensional arrays in forward mode

Coding Begins	
Week 1	***Description :*** Add support for differentiating w.r.t multi-dimensional arrays in the forward mode. This would involve modifying the "***ForwardModeVisitor::Derive"*** method. ***Deliverable :*** A working code snippet showing clad can differentiate the functions w.r.t multi-dimensional arrays.
Week 2 - Week 4	***Description :*** Add support for differentiating w.r.t pointers in reverse mode AD. This would involve extending the ***"ReverseModeVisitor"*** class to add support for differentiating w.r.t pointers. ***Deliverable :*** Working code snippet showing differentiating w.r.t pointers in reverse mode.
Week 5	***Description :*** ● Add demos for multi-dimensional array support in forward mode and pointer support in reverse mode AD. ● Fix any bugs, inefficient code. ***Deliverable :*** ● Demos for both the implementations. ● Fix bugs
Phase 1 Evaluations (Buffer Week)	
Week 7	***Description :*** ● Add support for differentiating w.r.t multi-dimensional arrays in reverse mode ***Deliverable :*** ● A working code snippet showing

	differentiation w.r.t a multidimensional array using ***"clad::gradient".***
Week 8	***Description :*** ● Fix bugs in the implementation if any and analyse the code for any bottlenecks and replace it with more efficient code. ***Deliverable :*** Fix any bugs in code
Week 9 - Week 11	***Description :*** ● Add demos showing differentiation w.r.t multi-dimensional arrays. ● Add an exhaustive set of test cases for all the features implemented. ● Add docs for differentiating w.r.t multi-dimensional arrays and pointers along with code snippets in the developer's documentation. ***Deliverable :*** ● Add demos demonstrating support for differentiation w.r.t multi-dimensional arrays. ● Add tests for differentiation w.r.t multi-dimensional arrays and pointers. ● Add documentation of the implemented features in the developer's documentation.
Final Evaluation (Buffer Week)	

Name and Contact Info :

Full Name - Rishabh Bali
Primary Email: rishabhsbali@gmail.com

Secondary Email (via my College): rsbali_b20@ce.vjti.ac.in
University: Veermata Jijabai Technological Institute, Mumbai
Ongoing Degree: Bachelor of Technology in Computer Engineering
CGPA : 8.53
Phone Number - 9821872320
Time-Zone: **IST** (UTC +5:30)
Profile: Github, Linkedin**,** Resume

Why me :

I am an undergraduate student in my third year at Veermata Jijabai Technological Institute (VJTI), Mumbai pursuing Computer Engineering. I have been familiar with Core Computer Science fundamentals for a long time , and have a great programming experience. I have completed many courses in my academic life and thus have considerable knowledge of Operating Systems, Compilers, Parallel Computing. Additionally we also had subjects like Linear Algebra and Differential Calculus, Discrete Mathematics denoting I have a strong mathematical background as well. I am an open source enthusiast who is well versed with C++ and Python. I am also fascinated by compiler optimizations. I also have some knowledge about LLVM, and the MLIR Framework. I also strive to write readable, modular and efficient code. I have successfully built clad and am currently exploring its codebase.

Experience with Open Source Projects :

I have been contributing to open source projects for a long time. In addition to contributing to projects like sktime and mlpack. I have also worked as an Open Source contributor at Buddy Compiler. Under the Open Source Promotion Plan (OSPP). I worked on adding Morphological Transformations to the DIP Dialect of buddy-mlir. The Digital image processing (DIP) dialect is created for the purpose of developing an MLIR backend for performing operations such as correlation, convolution, morphological processing. Our work on the DIP Dialect has also been accepted at **EuroLLVM 2023** Developer's meeting. Some

of my **contributions** can be found **here**. I have also added **benchmarks** for my implementation.

Contributions to Clad :

- I have raised the following issue relating to a minor correction in the Developer's docs.
- I have also worked on a PR to solve the above mentioned issue.
- I am also working on a PR to solve the following issue pertaining to removing dependency on the LastKnownGood*.txt files.

Availability :

- I do not have any commitments this summer and am ready to work for 30-35 hrs a week (for 12 weeks).
- I have my summer vacation from late May to September and I plan to complete a major part of the project in this period.
- I have my college examinations in October, but I plan to complete all the major work before the final deadline.
- I am ready to invest my weekends on this project as I find the idea to be very interesting.

Communication :

I believe that communication between mentor and mentee is a vital aspect of GSoC and to ensure that the status of the project is communicated properly, I will be undertaking the following steps:

I will publish a blog every week detailing:

- The work done for that week
- The problems faced by me during that week.
- Solutions to the problem faced.

I will also ensure that I give regular updates to the mentors (via gmail).

My Work Environment :

CPU : AMD Ryzen 7 4800H

GPU : NVIDIA GTX 1650 (4GB RAM)

Main Memory : 16GB

Secondary Storage : 512GB SSD

Operating System : Pop OS 22.04 LTS (Based on Ubuntu 22.04 LTS)