Clad - Clang plugin for Automatic Differentiation

Jack Qiu
Google Summer of Code 2019
Mentors: Aleksandr Efremov, Vassil Vassilev, Oksana Shadura
Automatic Differentiation - Forward Mode

- Differentiation is fixed w.r.t to a independent variable
- Breaks a function up into a list of sub-expressions/sequence of elementary operations
- Computes the derivative of each sub-expression recursively
- Implemented in Clad through clad::differentiate

\[
\begin{align*}
  z &= f(x_1, x_2) \\
  &= x_1 x_2 + \sin x_1 \\
  &= w_1 w_2 + \sin w_1 \\
  &= w_3 + w_4 \\
  &= w_5
\end{align*}
\]

\[
\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w_{n-1}} \frac{\partial w_{n-1}}{\partial x} = \frac{\partial y}{\partial w_{n-1}} \left( \frac{\partial w_{n-1}}{\partial w_{n-2}} \frac{\partial w_{n-2}}{\partial w_{n-3}} \frac{\partial w_{n-3}}{\partial x} \right) = \ldots
\]
Automatic Differentiation - Reverse Mode

- Differentiation is fixed w.r.t to the dependent variable
- We break function into sub-expressions, apply chain rule starting from the dependent variable
- Very effective for large no. of independent variables, but requires significant computation memory
- Implemented in Clad through clad::gradient

\[
\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w_1} \frac{\partial w_1}{\partial x} = \left( \frac{\partial y}{\partial w_2} \frac{\partial w_2}{\partial w_1} \right) \frac{\partial w_1}{\partial x} = \left( \left( \frac{\partial y}{\partial w_3} \frac{\partial w_3}{\partial w_2} \right) \frac{\partial w_2}{\partial w_1} \right) \frac{\partial w_1}{\partial x} = \ldots
\]
Hessians

- Square $n \times n$ matrix containing all second order partial derivatives w.r.t to all inputs
- Useful for optimisation problems and as a second derivative test

$$
H = \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{bmatrix}
$$
Hessians - How it is implemented

- Generated through using forward mode AD, then reverse mode AD
- Iteratively calculates each column of the Hessian at a time, which is encapsulated within a second-order partial derivative function
- Combines all of these helper functions that correspond to columns of a Hessian into a single Hessian function
- Encapsulated in Clad API through \texttt{clad::hessian}

\[
H = \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}
\end{bmatrix}.
\]
For more information, visit:
https://github.com/vgvassilev/clad