The Journal of Open Source Software

Mishra-Warren Autodiff: A Generalized Input Automatic Differentiation library in C++

Neil Warren[*1], **Aakash Mishra**[†2], **David Sondak**[‡3], **and Andrew Kirby**[§4]

1 Software Engineering Master's Degree Candidate, Harvard University, Extension School **2** Computer Science Major, Harvard College **3** Lecturer on Computational Science, Harvard Institute for Applied Computational Science **4** Post-Doctoral Associate, MIT Lincoln Laboratory

DOI: DOIunavailable

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Reviewers:

- @Pending Reviewers

Submitted: N/A
Published: N/A

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

Summary

From mathematical optimization to the training of neural networks, the need to efficiently compute derivatives of complex functions of tens or even thousands of variables and operations have become ubiquitous in the field of machine learning. (Baydin et al., 2018) Given the computational drawbacks of performing symbolic differentiation of complex functions, which quickly become intractable for even the most sophisticated computer algebra systems, automatic differentiation has become the preferred method of derivative computation. Notably, several popular open source machine learning libraries include automatic differentiation functionality: Pytorch (Paszke et al., 2017), (PyTorch.org, 2020); Tensorflow (TensorFlow.org, 2020). While powerful and extensible, these libraries are limited in the scope and form of the input that they can accept. In particular, users of these libraries use computer source code (often Python) to input the function to be differentiated or write source code to derive the internal representation of the functions. For functions derived outside of the Python environment, or from an otherwise accessible computational environment, function definition and data input is a largely manual process.

The Mishra-Warren Autodiff library utilizes forms of generalized input (i.e., non-specific to the computational environment), such as scientific documents and text strings. Accepting more generalized forms of input allows for the efficient automation of automatic differentiation for functions derived in varied computational environments. This is particularly useful for large functions of tens or even thousands of variables, for example, as provided by output in text form from documents or as output from libraries written in other languages with incompatible data types.

Statement of need

In many applications, scientists may use a software library that lacks built-in efficient automatic differentiation functionality to generate complex function with many variables and need to efficiently calculate the derivative or a series of derivatives of the function. For large functions of tens or even thousands of variables the data input process for providing that function to a standard machine learning library such as Pytorch or Tensorflow would be exceedingly time consuming and prone to error. Thus, there is a need for an automatic differentiation library that can read in the function to be evaluated in a generalized form, such as a raw text, that can be easily read out to a file by a software library or provided in a scientific document.

Mishra-Warren Autodiff is a C++ library that applies Dijkstra's shunting yard algorithm (Boysen et al., 2012) to pre-process textual function inputs along with point vectors and derivative

*Footnote description.
†Footnote description.
‡Footnote description.
§Footnote description.

seed vectors, and then applies forward and reverse mode automatic differentiation algorithms to develop a well-formed Jacobian matrix. In contrast, most existing popular automatic differentiation libraries take functional input in the form of a computer program's source code (often Python) utilizing specialized data structures and previously declared variables and constants. (TensorFlow.org, 2020) This makes the process of applying automatic differentiation to functions from textual sources a manual task. Mishra-Warren Autodiff accepts raw textual input, e.g., "f(x0,y0) = x0 * cos(y0)," from the command line or a text file, such as a scientific document.

Features

Mishra-Warren Autodiff takes as input functional descriptions and provides as output a complete well-formed Jacobian matrix for the function. The user may optionally select either forward mode or reverse mode automatic differentiation, the relative advantages and disadvantages of which have discussed at length in the literature. (Margossian, 2018) (Baydin et al., 2018)

As shown in **??**, the Mishra-Warren Autodiff shunting yard pre-processing implementation achieves approximately O(n) time complexity over the number of generated tokens (approximately proportional to the number of elementary functions in the input function). This was confirmed using a standard laptop with an Intel® Core™ i7-8550U CPU running 100 averaged trials with functions of up to 524,288 elementary operations and independent variables. Independent of the functional complexity, given that variable names given in text (e.g., "x0" to "x524287") grow in length, parsing for variable names in the tokenizer causes the shape of the curve to become more quadratic as the input size is increases.

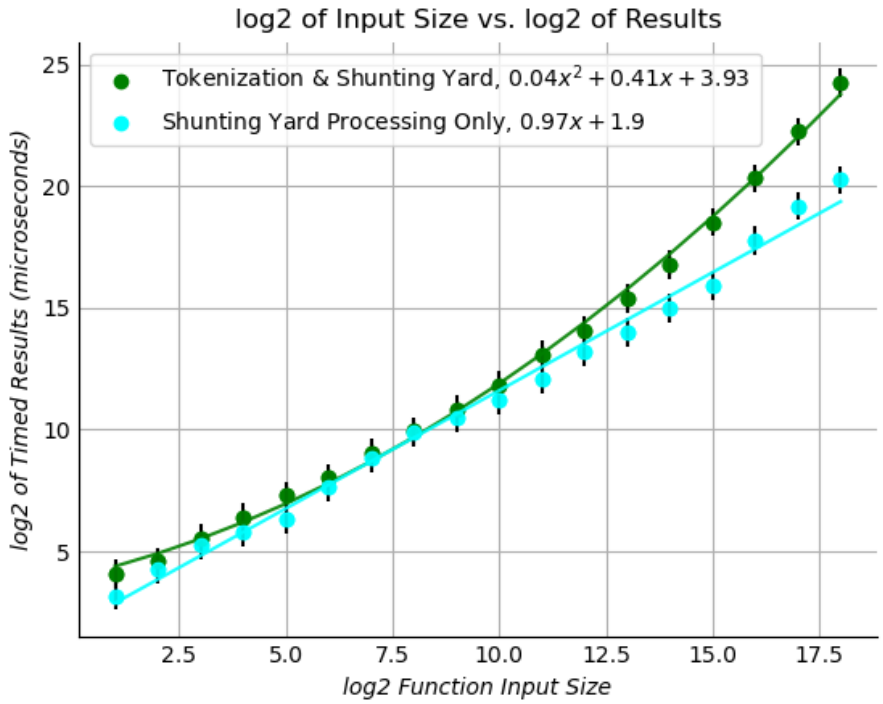

Figure 1: Parsing tokenization scaling.

Given the algorithmic constraints of forward mode automatic differentiation, which computes

partial derivatives for each independent variable at each computational node, reverse mode provides for more efficient computation of derivatives for functions with a large numbers of independent variables. Reverse mode, in contrast, requires the computation of an adjoint tree from a first pass through the function tokens, which adds a small amount of computational overhead and thus is not optimal for functions of on a few variables. An exemplary description of computational time complexity for the parsing and evaluation of the Jacobian for functions of 1 to 2^{18} variables is provided below.

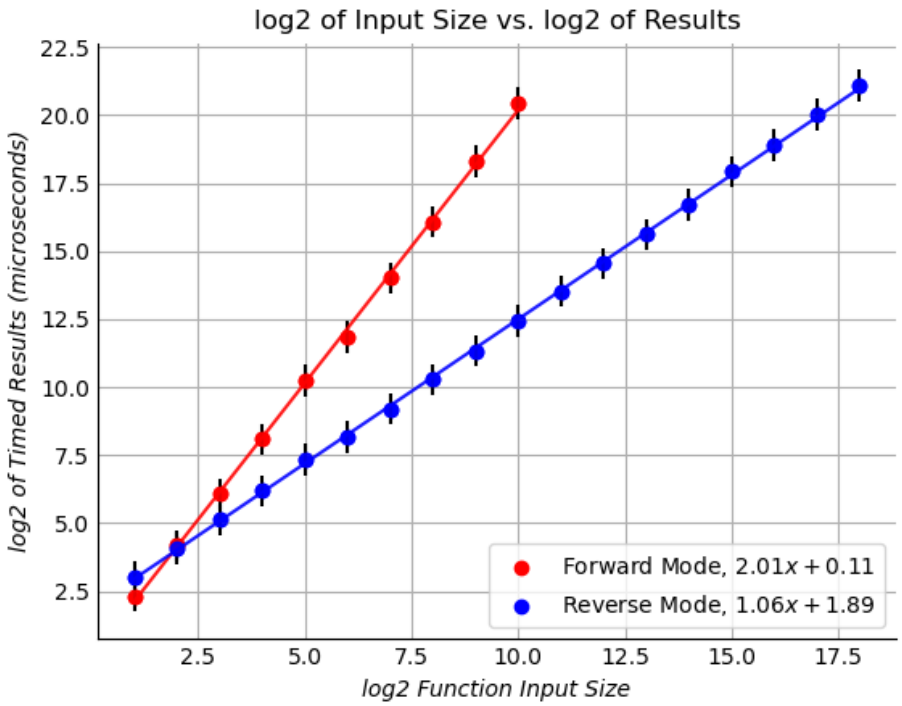

Figure 2: Forward vs. Reverse Mode.

Acknowledgements

Mishra-Warren Autodiff was developed as an extension of a group project developed in the course "Systems Development for Computational Science," CS-107, at Harvard University in the Fall of 2020 under the instruction of Dr. David Sondak, Harvard Institute for Applied Computational Science, and Dr. Andrew Kirby, Post-doctoral associate at MIT Lincoln Laboratory, and in collaboration with Leo Landau and Samson Negassi.

References

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., & Siskind, J. M. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, *18*(153), 1–43. http://jmlr.org/papers/v18/17-468.html

Boysen, N., Fliedner, M., Jaehn, F., & Pesch, E. (2012). Shunting yard operations: Theoretical aspects and applications. *European Journal of Operational Research*, *220*(1), 1–14.

Margossian, C. C. (2018). A review of automatic differentiation and its efficient implementation. *CoRR*, *abs/1811.05031*. http://arxiv.org/abs/1811.05031

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. (2017). *Automatic differentiation in PyTorch*.

PyTorch.org. (2020). *A GENTLE INTRODUCTION TO TORCH.AUTOGRAD*.

TensorFlow.org. (2020). *Introduction to Gradients and Automatic Differentiation*.