The Simple Essence of Automatic Differentiation

CONAL ELLIOTT, Target, USA

Automatic differentiation (AD) in reverse mode (RAD) is a central component of deep learning and other uses of large-scale optimization. Commonly used RAD algorithms such as backpropagation, however, are complex and stateful, hindering deep understanding, improvement, and parallel execution. This paper develops a simple, generalized AD algorithm calculated from a simple, natural specification. The general algorithm is then specialized by varying the representation of derivatives. In particular, applying well-known constructions to a naive representation yields two RAD algorithms that are far simpler than previously known. In contrast to commonly used RAD implementations, the algorithms defined here involve no graphs, tapes, variables, partial derivatives, or mutation. They are inherently parallel-friendly, correct by construction, and usable directly from an existing programming language with no need for new data types or programming style, thanks to use of an AD-agnostic compiler plugin.

CCS Concepts: • **Mathematics of computing** → **Differential calculus**; • **Theory of computation** → *Program reasoning*; *Program specifications*;

Additional Key Words and Phrases: automatic differentiation, program calculation, category theory

ACM Reference Format:
Conal Elliott. 2018. The Simple Essence of Automatic Differentiation. *Proc. ACM Program. Lang.* 2, ICFP, Article 70 (September 2018), 29 pages. https://doi.org/10.1145/3236765

1 INTRODUCTION

Accurate, efficient, and reliable computation of derivatives has become increasingly important over the last several years, thanks in large part to the successful use of *backpropagation* in machine learning, including multi-layer neural networks, also known as "deep learning" [Goodfellow et al. 2016; Lecun et al. 2015]. Backpropagation is a specialization and independent invention of the *reverse mode* of automatic differentiation (AD) and is used to tune a parametric model to closely match observed data, using *gradient descent* (or *stochastic* gradient descent). Machine learning and other gradient-based optimization problems typically rely on derivatives of functions with very high dimensional domains and a scalar codomain—exactly the conditions under which reverse-mode AD is much more efficient than forward-mode AD (by a factor proportional to the domain dimension). Unfortunately, while forward-mode AD (FAD) is easily understood and implemented, reverse-mode AD (RAD) and backpropagation have had much more complicated explanations and implementations, involving mutation, graph construction and traversal, and "tapes" (sequences of reified, interpretable assignments, also called "traces" or "Wengert lists"). Mutation, while motivated by efficiency concerns, makes parallel execution difficult and so undermines efficiency as well. Construction and interpretation (or compilation) of graphs and tapes also add execution overhead. The importance of RAD makes its current complicated and bulky implementations especially problematic. The increasingly large machine learning (and other optimization) problems being solved with RAD (usually via backpropagation) suggest the need to find more streamlined, efficient

Author's address: Conal Elliott, conal@conal.net, Target, USA.

2475-1421/2018/9-ART70
https://doi.org/10.1145/3236765