Implicit Differentiation

Justin Chiu
Cornell Tech
jtc257@cornell.edu

July 7, 2021

Abstract

Gradient-based learning forms the foundation of modern machine learning, and automatic differentiation allows ML practitioners to easily compute the gradients necessary for learning. While automatic differentiation only costs a constant multiple of the time and space required to evaluate a function, it has its limitations. In particular, when evaluating a function itself is expensive, the direct application of automatic differentiation is infeasible. In this report, we review the implicit function theorem (IFT) and its use in reducing the cost of computing gradients in scenarios where function evaluation is expensive, focusing on the application of implicit differentiation to variational inference.

1 Introduction

Gradient-based learning underpins many of the recent successes in machine learning, particularly advances involving neural networks. The key to the success of gradient-based methods is automatic differentiation (AD), which has greatly increased the development speed of machine learning research by allowing practitioners to circumvent the error-prone and time-consuming process of computing gradients manually. AD operates by reducing functions into compositions of atomic operations, for which we have a library of derivatives for, and composing those derivatives via the chain rule. (introduce the representation of functions as evaluation procedures / computational graphs, following Griewank and Walther [2008]) While efficient relative to the evaluation of the function in question, taking only a multiplicative constant longer than the evaluation itself, this may be prohibitively expensive if the original function evaluation itself is costly. An example of this is if the function takes the form of an unrolled loop, a common artifact of iterative methods. As naive AD requires storing all of the intermediate values at each point, storing the output of all computations at every iteration of a loop can quickly become infeasible due to memory limitations.

There are a variety of methods for overcoming the space limitations of AD, of which we only mention three: checkpointing, reversible computation, and implicit differentiation. A first method, checkpointing, improves space complexity at the cost of time. Rather than storing all intermediate computation, checkpointing instead recomputes values when needed. This can result in a large slowdown, and also requires careful choosing of which computationals subgraphs to checkpoint. A

second method is an improvement upon checkpointing, called reversible computation [Maclaurin et al., 2015, Gomez et al., 2017], which improves space complexity at the cost of expressivity, but not speed. Reversible computation ensures that the gradient with respect to input depends only on the output, allowing the input to be discarded during function evaluation. This is typically accomplished by ensuring that the input is easily reconstructed from the output, restricting the expressivity of layers. A third method is implicit differentiation, which improves space complexity at the cost of BLANK. Implicit differentiation relies on the implicit function theorem (IFT) [], which gives conditions under which derivatives can be computed independent of intermediate computation. In this report, we will cover the implicit function theorem (IFT), discuss the intuition behind implicit differentiation, and apply it to reducing the computational cost of taking derivatives through variational inference.

Bilevel Optimization (Need to be careful and double check this sentence, since IFT is probably more general than just bilevel opt) Informally, applications of implicit differentiation can commonly be formulated as bilevel optimization problems, where, for every iteration when solving an outer optimization problem, we must additionally solve an inner optimization problem. Some examples of bilevel optimization problems are hyperparameter optimization, metalearning, and variational inference.

Hyperparameter optimization formulates hyperparameter tuning, such as the shrinkage penalty in LASSO, as a bilevel optimization problem by computing gradients wrt the penalty through the entire learning procedure of the linear model [Lorraine et al., 2019]. (Other works on hyperparam opt [Maclaurin et al., 2015, Bertrand et al., 2020]) Similarly, metalearning learns the parameters of a model such that is the model is able to quickly be adapted to a new task via gradient descent [Finn et al., 2017, Rajeswaran et al., 2019]. This is accomplished by differentiating through the learning procedure of each new task. Finally, a variant of variational inference follows a very similar format: semi-amortized variational inference (SAVI) aims to learn a model that is able to initialize variational parameters [Kim et al., 2018]. This is also accomplished by differentiating through the iterative optimization procedure applied to the variational parameters during inference. (Other VI papers [Wainwright and Jordan, 2008, Johnson et al., 2017])

There is also work on expressing individual layers of a neural network declaratively as the solution of an optimization problem [Amos and Kolter, 2017, Agrawal et al., 2019, Gould et al., 2019]. This also falls under the umbrella of bilevel optimization, as we have both the outer training loop for the whole model and the inner optimization loop for each optimization layer. (reword)

2 The Implicit Function Theorem

References

A. Agrawal, B. Amos, S. T. Barratt, S. P. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. *CoRR*, abs/1910.12430, 2019. URL `http://arxiv.org/abs/1910.12430`.

B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. *CoRR*, abs/1703.00443, 2017. URL http://arxiv.org/abs/1703.00443.

Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of lasso-type models for hyperparameter optimization, 2020.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. URL http://arxiv.org/abs/1703.03400.

A. N. Gomez, M. Ren, R. Urtasun, and R. B. Grosse. The reversible residual network: Backpropagation without storing activations. *CoRR*, abs/1707.04585, 2017. URL http://arxiv.org/abs/1707.04585.

S. Gould, R. Hartley, and D. Campbell. Deep declarative networks: A new hope. *CoRR*, abs/1909.04866, 2019. URL http://arxiv.org/abs/1909.04866.

A. Griewank and A. Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, 2nd edition, 2008.

M. J. Johnson, D. Duvenaud, A. B. Wiltschko, S. R. Datta, and R. P. Adams. Composing graphical models with neural networks for structured representations and fast inference, 2017.

Y. Kim, S. Wiseman, A. C. Miller, D. Sontag, and A. M. Rush. Semi-amortized variational autoencoders, 2018.

J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. *CoRR*, abs/1911.02590, 2019. URL http://arxiv.org/abs/1911.02590.

D. Maclaurin, D. Duvenaud, and R. P. Adams. Gradient-based hyperparameter optimization through reversible learning, 2015.

A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. *CoRR*, abs/1909.04630, 2019. URL http://arxiv.org/abs/1909.04630.

M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 01 2008. doi: 10.1561/2200000001.

A Example Appendix

Neural ODEs use reversibility.