Nonsmooth Implicit Differentation for Machine Learning

Antonio Silveti-Falls
Toulouse School of Economics
tonys.falls@gmail.com

Introduction Many problems in machine learning can be solved efficiently by taking advantage of implicit differentiation, from hyperparameter tuning [1] to training neural networks with implicitly defined layers [3, 2]. The key ingredient to applying implicit differentiation is the implicit function theorem which guarantees the existence of an implicit function and its differentiability, with a calculus for the implicit gradient. A bottleneck for extending such methods in practice is the lack of smoothness present in many machine learning problems. Although there is a rich literature on nonsmooth implicit function theorems already, the focus has primarily been on proving the existence and regularity of implicit functions rather than on developing a practical calculus.

Main Result We construct a theory of implicit differentiation for path differentiable functions [4] with a flexible calculus that allows one to compute implicit gradients using the analogous formulas from the smooth setting, in a way that is compatible with backpropagation and algorithmic differentiation. Path differentiable functions were studied in [4] as a subset of locally Lipschitz functions which admit a *conservative Jacobian*, denoted $\mathcal{J}_F$ for a function F. Our main contribution is the following theorem.

Theorem. Let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ be path differentiable and $(\hat{x}, \hat{y}) \in \mathbb{R}^n \times \mathbb{R}^m$ be such that $F(\hat{x}, \hat{y}) = 0$. Assume $\mathcal{J}_F(\hat{x}, \hat{y})$ is convex and $\forall [A\ B] \in \mathcal{J}_F(\hat{x}, \hat{y})$, B is invertible. Then $\exists U \subset \mathbb{R}^n$ a neighborhood of $\hat{x}$ and a path differentiable function G such that

$$\forall x \in U \qquad F(x, G(x)) = 0.$$

A conservative Jacobian of G can be computed from the formula

$$\mathcal{J}_G(x) = \left\{-B^{-1}A : [A\ B] \in D_F(x, G(x))\right\}.$$

Applications With the previous theorem and the framework of conservative Jacobians, we are able to prove almost sure convergence guarantees for training neural networks with implicitly defined layers. These are networks with layer outputs defined as fixed points to an equation [2] or solutions to an optimization problem [3].

We also examine hyperparameter tuning for the LASSO. The problem of choosing the best weight λ for the LASSO problem can be formulated as a bilevel optimization problem:

$$\min_{\lambda \in \mathbb{R}} C(\hat{\beta}(\lambda)) \quad \text{such that} \quad \hat{\beta}(\lambda) \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} \frac{1}{2} \|X\beta - y\|_2^2 + \lambda \|\beta\|_1$$

where C is some measure of task performance, e.g. the cross validation loss, the holdout loss, the stein unbiased risk estimate, etc. By writing the optimality condition as a fixed point equation, we can apply our theorem to compute a conservative Jacobian for the solution $\hat{\beta}(\lambda)$.

Joint work with: Jérôme Bolte, Tam Le, and Edouard Pauwels.

References

[1] Pedregosa, Fabian Hyperparameter optimization with approximate gradient *International conference on machine learning* 2016

[2] Bai, Shaojie and Kolter, J Zico and Koltun, Vladlen. Deep equilibrium models *Advances in Neural Information Processing Systems* 2019

[3] Agrawal, Akshay and Amos, Brandon and Barratt, Shane and Boyd, Stephen and Diamond, Steven and Kolter, J Zico Differentiable convex optimization layers *Advances in neural information processing systems* 2019

[4] Bolte, Jérôme and Pauwels, Edouard Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning *Mathematical Programming* 2021