Approximate Automatic Differentiation through the IFT

Justin Chiu
Cornell Tech
jtc257@cornell.edu

September 25, 2021

Abstract

Gradient-based learning forms the foundation of modern machine learning, and automatic differentiation allows ML practitioners to easily compute gradients. While a single application of forward or reverse mode automatic differentiation to a function is guaranteed to be within a constant factor of the original function's runtime, this applies only to exact differentiation. Inspired by randomized automatic differentiation [Oktay et al., 2020], we explore trading off accuracy for speed and/or memory by introducing approximations into automatic differentiation.

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 greatly increases 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. The underlying concept behind AD is that a program's execution trace is a valid and useful representation of a function [Griewank and Walther, 2008], where the execution trace is a history of all the computation performed during the evaluation of a program used to express a function.

Introduce space / time complexity

2 Speedups and Spacedowns

In this section we will revisit the softmax problem and experiment with whether the derivative can be approximated more efficiently in either time or space. First we will review the efficient vector-Jacobian product (VJP), then attempt to apply different approximations in the IFT to see if we can find a sublinear algorithm.

Recall softmax,

$$\text{softmax}(\theta)_i = \frac{\exp(\theta_i)}{\sum_j \exp(\theta_j)},$$

and the softmax problem from Eqn. **??**,

$$\begin{aligned} \text{maximize} \quad & z^\top \theta + H(z) \\ \text{subject to} \quad & z^\top \mathbf{1} = 1 \\ & z_i \geq 0, \forall i. \end{aligned} \tag{1}$$

We would like to compute $\frac{dz}{d\theta} = \text{diag}(z) - zz^\top$. The exact directional derivative (vjp) can be computed in $O(n)$ operations, given by $v^\top \frac{dz}{d\theta} = z \cdot (v - (z^\top v)\mathbf{1}_n)$ (this can be computed in $2n$ multiplications and $3n$ additions). We will explore whether using vector-inverse Hessian products can result in fewer operations.

IFT and Neumann Approximation Our first attempt will use a combination of the IFT and a Neumann approximation. Recall that the IFT tells us

$$\frac{dx}{d\theta} = \frac{dx^*(\theta)}{d\theta} = -\left[\frac{dF(\theta, x)}{dx}\right]^{-1} \frac{dF(\theta, x)}{d\theta},$$

where

$$\frac{dF(\theta, x)}{d\theta} = \begin{bmatrix} I_{n\times n} \\ \mathbf{0}_{(n+1)\times(n+1)} \end{bmatrix}$$

and

$$\left[\frac{dF(\theta, x)}{dx}\right]^{-1} = \begin{bmatrix} -\text{diag}(z)^{-1} & \mathbf{1}_n & I_{n\times n} \\ u\mathbf{1}_n^\top & 0 & \mathbf{0}_n^\top \\ \mathbf{0}_{n\times n} & \mathbf{0}_n & \text{diag}(z) \end{bmatrix}^{-1}.$$

Recall that we restrict our attention to $z \succ 0$, so $v = 0$ by complementary slackness. The Neumann approximation is given by $X^{-1} = \sum_{k=0}^{\infty}(I - X)^k$. We will use a truncated (at $k = 1$) Neumann approximation

$$\begin{aligned} \left[\frac{dF(\theta, x)}{dx}\right]^{-1} &\approx 2I_{n\times n} - \frac{dF(\theta, x)}{dx} \\ &= \begin{bmatrix} 2I_{n\times n} + \text{diag}(z)^{-1} & -\mathbf{1}_n & -I_{n\times n} \\ -u\mathbf{1}_n^\top & 2 & \mathbf{0}_n^\top \\ \mathbf{0}_{n\times n} & \mathbf{0}_n & 2I_{n\times n} - \text{diag}(z) \end{bmatrix}, \end{aligned}$$

yielding

$$\frac{dz}{d\theta} \approx -2I - \text{diag}(z)^{-1}.$$

Computing the vector-Jacobian product (vjp), we have

$$v^\top \frac{dz}{d\theta} \approx -v \circ (2\mathbf{1}_n + z^{-1}).$$

While this is still $O(n)$, this does potentically take fewer operations than the exact derivative: n multiplications, n reciprocals, and n additions if the constant terms are pre-computed.

This approximation is quite poor in practice. We check if the higher-order terms of the Neumann series result in a better approximation.

We first note that we can solve for the Lagrange multiplier u. By setting $\frac{d\mathcal{L}(\theta,z,u,v)}{d(z,u)} = 0$ and noting that $z_i = \text{softmax}(\theta)_i = \exp(\theta_i)/\sum_j \exp(\theta_j)$, we have that $u = \log \sum_i \exp(\theta_i) + 1 = \text{LSE}(\theta) + 1$.

Next, we examine the product $(I - \frac{dF(\theta,x)}{dx})^k$. Let us start with $n = 2$ and $k = 2$:

$$\left(I - \frac{dF(\theta,x)}{dx}\right)^2 = \begin{bmatrix} \text{diag}(1+z^{-1}) & -\mathbf{1}_n & -I_n \\ -u\mathbf{1}_n^\top & 1 & \mathbf{0}_n^\top \\ \mathbf{0}_{n\times n} & 0 & \text{diag}(1-z) \end{bmatrix}^2$$

$$= \begin{bmatrix} 1+1/z_1 & 0 & -1 & -1 & 0 \\ 0 & 1+1/z_2 & -1 & 0 & -1 \\ -u & -u & 1 & 0 & 0 \\ 0 & 0 & 0 & 1-z_1 & 0 \\ 0 & 0 & 0 & 0 & 1-z_2 \end{bmatrix}^2$$

$$= \begin{bmatrix} u + (1+1/z_1)^2 & u & -1/z_1 - 2 & z_1 - 1/z_1 & 0 \\ u & u + (1+1/z_2)^2 & -1/z_2 - 2 & 0 & z_2 - 1/z_2 - 2 \\ -u(1/z_1+2) & -u(1/z_2+2) & 2u+1 & u & u \\ 0 & 0 & 0 & (1-z_1)^2 & 0 \\ 0 & 0 & 0 & 0 & (1-z_2)^2 \end{bmatrix}$$

Unfortunately, this is both unstable due to division by z_1 and z_2, and also requires more operations than the first term $I - X$ (with $X = \frac{dF}{dx}$ for brevity) without a clear way of simplifying the sum $\sum_k (I - X)^k$. Thus, we find that the IFT and Neumann series approximation does not provide benefit over the direct derivative for softmax. Only the first-order approximation has fewer operations but large error.

3 Structured Softmax

We now apply the IFT to differentiating through the marginals of structured distributions.

Structured distributions extend softmax in fundamental way: Softmax scores each configuration globally, requiring an exponential amount of computation to compute the partition function. Structured distributions instead break down the scoring of configurations into local scoring functions, often resulting in the partition function being computable in polynomial time. In particular, we focus on the simple case of a first-order linear-chain CRF. Give details later.

Similar to the softmax case, we can compute both the marginals z and the gradient of the marginals wrt the parameters $\frac{dz}{d\theta}$ exactly in time polynomial in n.

References

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

D. Oktay, N. McGreivy, J. Aduol, A. Beatson, and R. P. Adams. Randomized automatic differentiation. *CoRR*, abs/2007.10412, 2020. URL `https://arxiv.org/abs/2007.10412`.

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 Properties of Exponential Family Distributions

Exponential family distributions are widely used to parameterize graphical models [Wainwright and Jordan, 2008]. The classification of a distribution as exponential family simply says that we can write their joint distributions with the particular factorization given above, which can then be used to derive useful properties common to all exponential family distributions. These properties, made clearer through the exponential family abstraction, are then integral for performing inference. In particular, gradient estimation can be written as mapping from canonical to mean parameters.

A.1 Canonical Parameters

Exponential family distributions take the form

$$p(x;\theta) = h(x)\exp(\theta^\top \phi(x) - A(\theta)), \tag{2}$$

where $h : \mathcal{X} \to \mathbb{R}_+$ is the base measure, $\theta \in \mathbb{R}^d$ the canonical parameters, $\phi : \mathcal{X} \to \mathbb{R}^d$ the sufficient statistics[1], and $A : \mathbb{R}^d \to \mathbb{R}$ the log partition function. Intuitively, the canonical parameters θ score configurations $x \in \mathcal{X}$, with representation $\phi(x)$. The log partition function $A(\theta) = \log \int_{\mathcal{X}} h(x)\exp(\theta^\top \phi(x))$ is used to ensure normaliziation of the probability density, i.e. $\int_{\mathcal{X}} p(x) = 1$. The base measure $h : \mathcal{X} \to \mathbb{R}$ is also used to ensure proper normalization, but is only a function of x. It can also be used to enforce hard constraints in models by giving 0 mass to some configurations x.

We give three examples of exponential family distributions: softmax, hidden Markov models (HMMs) and linear chain conditional random fields (CRFs).

Softmax Softmax gives a discrete distribution over n items, where $x \in [1, \dots, n]$. The probability mass function is then $p(x;\theta) = \exp(\theta^\top x - A(\theta))$, where $A(\theta) = \log \sum_i \exp(\theta_i) = \log \exp(\theta)^\top \mathbf{1}_n$. Thus, the sufficient statistics are $\phi(x)_i = 1(x_i)$ for each $i \in [n]$ and the base measure constant $h(x) = 1$.

[1] Potential functions usually refer to particular pairs of canonical parameter and sufficient statistic products $\theta_\alpha \phi_a lpha(x)$ as a function of x.

Hidden Markov Models (HMMs) Hidden Markov models (HMMs) give a discrete distribution over sequences $x = (x_1, \ldots, x_T)$, where $x_t \in [X]$, and latent states $z = (z_1, \ldots, z_T)$, where $z_t \in [Z]$. HMMs have the joint distribution $p(x, z) = \prod_t p(x_t \mid z_t)p(z_t \mid z_{t-1})$, with distinct start state $z_0 = \epsilon$. We assume each condition distribution is parameterized using softmax, and the HMM has canonical parameters $\theta = (O, S)$ for the emission and transitions respectively. The emissions are locally normalized, given by $p(x_t \mid z_t) = O1_{z_t,x_t}$ where $O \in [0,1]^{ZX}$, $1_{z_t,x_t} \in \mathbb{R}^{ZX}$ is a one-hot vector. Local normalization implies $\sum_x O_{z,x} = 1$ for a given z. The transitions are also locally normalized, given by $p(z_t \mid z_{t-1}) = S1_{z_{t-1},z_t}$, where $S \in [0,1]^{Z^2}$, $1_{z_{t-1},z_t} \in \{0,1\}^{Z^2}$ is a one-hot vector. Both O and S are vector representations of the observation and state transition matrices respectively. The sufficient statistics are $\{1(x_t, z_t)\}_{x_t,z_t}$ and $\{1(z_{t-1}), 1(z_t)\}_{z_{t-1},z_t}$ for all t, x_t, z_{t-1}, z_t. The base measure is constant $h(x, z) = 1$.

Linear Chain Conditional Random Fields (CRFs) The linear chain CRF has states $x = (x_1, \ldots, x_T)$, where $x_t \in [Z]$, and joint distribution

$$p(x) = \frac{\prod_t f(z_t; \theta) g(z_{t-1}, z_t; \theta)}{\exp(A(\theta))} = \exp(\theta^\top \phi(x) - A(\theta)),$$

where $f : \mathcal{X} \to \mathbb{R}_+, g : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ are unary and binary potentials respectively. These potentials are the most common representation of CRFs. In the exponential family representation, the canonical parameters $\theta \in \mathbb{R}^{(Z+Z^2)T}$ are used to score unary and binary terms, given by the sufficient statistics $\phi(x)_{t,1,i} = 1(z_t = i)$ and $\phi(x)_{t,2,ij} = 1(z_{t-1} = i, z_t = j)$ for t, i, j. The base measure is constant $h(x) = 1$.

A.2 Mean Parameters

The mean parameters of exponential family distributions are the expected sufficient statistics: $\mu = \mathbb{E}_{p(x)}[\phi(x)]$. The space of all possible marginals, obtained by varying the distribution $p(x)$, is called the mean parameter space

$$\mathcal{M} = \{\mu \mid \exists p \text{ s.t. } \mathbb{E}_p[\phi(x)] = \mu\}.$$

For the particular case of discrete graphical models, which is of particular interest, we call $\mathcal{M}$ the marginal polytope. In particular, for discrete models, we have

$$\mathcal{M} = \left\{ \mu | \sum_x p(x)\phi(x) = \mu, \sum_x p(x) = 1 \right\},$$

expressed as a finitely generated polyhedra. By the Minkowski-Weyl theorem, we can equivalently express the marginal polytope as a constrained polyhedra, i.e. the convex hull of the sufficient statistics:

$$\mathcal{M} = \text{conv}(\phi(x), x \in \mathcal{X}).$$

This representation is more useful in variational optimization problems, as it translates directly to a set of constraints.

A.3 Forward and Backward Mappings from Canonical to Mean Parameters

The forward mapping $\theta \mapsto \mu$ is marginal inference in CRFs, while the backward mapping $\mu \mapsto \theta$ corresponds to learning. The log partition function $A(\theta)$ is key to both of these mappings:

Forward $\nabla A(\theta) = \mu$

Backward $\theta \in \nabla A^*(\mu)$

where $A^*(\mu)$ denotes the conjugate dual function of A. When the mean parameters are in the interior of the mean parameter space (henceforth referred to as the marginal polytope, as we focus on discrete graphical models) $\mu \in \mathcal{M}^\circ$, the conjugate dual function $A^*(\mu) = -H(p_{\theta(\mu)})$ More precise definition following Wainwright and Jordan [2008] Thm. 3.4.

B Graphical Models

Neural ODEs use reversibility.