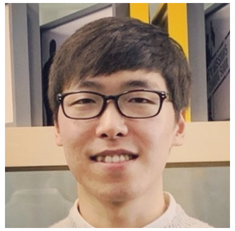
On the Correctness of Automatic Differentiation for Neural Networks with Machine-Representable Parameters



Wonyeol Lee[1]   Sejun Park[2]   Alex Aiken[1]

[1]Stanford University, USA   [2]Korea University, South Korea

ICML 2023

Automatic Differentiation

- Automatic differentiation (AD)[1] refers to various algorithms for computing the derivative

$$\mathcal{D}P(x) \in \mathbb{R}^{m \times n} \quad \text{(when it exists)}$$

  of a program $P : \mathbb{R}^n \to \mathbb{R}^m$ at an input $x \in \mathbb{R}^n$.

[1]Our paper focuses on two popular modes: forward-mode AD and reverse-mode AD (which includes backprop).

Automatic Differentiation

- Automatic differentiation (AD)[1] refers to various algorithms for computing the derivative

$$\mathcal{D}P(x) \in \mathbb{R}^{m \times n} \quad \text{(when it exists)}$$

  of a program $P : \mathbb{R}^n \to \mathbb{R}^m$ at an input $x \in \mathbb{R}^n$.

- Backpropagation is an instance of AD widely used in ML.

TensorFlow PyTorch …

[1]Our paper focuses on two popular modes: forward-mode AD and reverse-mode AD (which includes backprop).

Correctness of AD

fully connected, convolution, softmax, …

- If P consists of differentiable functions, then

$$\mathcal{D}P(x) \text{ exists} \quad \text{and} \quad \mathcal{D}^{\mathrm{AD}}P(x) = \mathcal{D}P(x) \qquad \text{for all } x \in \mathbb{R}^n.$$

output of AD on P at x

Correctness of AD

- If P consists of differentiable functions, then

$\mathcal{D}P(x)$ exists and $\mathcal{D}^{\mathrm{AD}}P(x) = \mathcal{D}P(x)$ for all $x \in \mathbb{R}^n$.

ReLU, max, abs, …

- If P uses non-differentiable functions, then

$\mathcal{D}P(x)$ might not exist or $\mathcal{D}^{\mathrm{AD}}P(x) \neq \mathcal{D}P(x)$ for some $x \in \mathbb{R}^n$.

Correctness of AD

- If P consists of differentiable functions, then

$$\mathcal{D}P(x) \text{ exists} \quad \text{and} \quad \mathcal{D}^{\mathrm{AD}}P(x) = \mathcal{D}P(x) \qquad \text{for all } x \in \mathbb{R}^n.$$

Prior work [Bolte+20, Lee+20, Huot+23, …]

"piecewise analytic"

include ReLU, max, abs, …

- If P uses ~~non-differentiable~~ functions, then

$$\mathcal{D}P(x) \text{ might not exist} \quad \text{or} \quad \mathcal{D}^{\mathrm{AD}}P(x) \neq \mathcal{D}P(x) \qquad \cancel{\text{for some}}\ x \in \mathbb{R}^n.$$

only for measure zero
(i.e., negligible)

Limitations of Prior Work

- In practice, inputs are not reals, but machine-representable numbers (e.g., floats).
- The set of machine-representable numbers $\mathbb{M}$ is countable, so has measure zero in $\mathbb{R}$.

“piecewise analytic”

- If P uses ~~non-differentiable~~ functions, then

$$\mathcal{D}P(x) \text{ might not exist} \qquad \text{or} \qquad \mathcal{D}^{\mathrm{AD}}P(x) \neq \mathcal{D}P(x) \qquad \underbrace{\cancel{\text{for some}}}_{\substack{\text{only for measure zero} \\ \text{(i.e., negligible)}}}\ x \in \mathbb{R}^n.$$

Limitations of Prior Work

- In practice, inputs are not reals, but machine-representable numbers (e.g., floats).
- The set of machine-representable numbers $\mathbb{M}$ is countable, so has measure zero in $\mathbb{R}$.

AD can be incorrect for all $x \in \mathbb{M}^n$ and this is indeed possible!

E.g., for $P = \frac{1}{|\mathbb{M}|} \sum_{c \in \mathbb{M}} \left(\mathrm{ReLU}(x - c) - \mathrm{ReLU}(-x + c) \right)$,

$$\mathcal{D}^{\mathrm{AD}} P(x) \neq \mathcal{D} P(x) \quad \text{for all } x \in \mathbb{M}.$$

Our Goal

Study the correctness of AD when inputs are machine-representable numbers.

- We focus on programs $P : \mathbb{R}^n \to \mathbb{R}^m$ that represent neural networks:

$$\underset{\substack{\uparrow \\ \text{parameters of a network}}}{w} \quad \longmapsto \quad P(w).$$

Our Goal

Study the correctness of AD when inputs are machine-representable numbers.

- We focus on programs $P : \mathbb{R}^n \to \mathbb{R}^m$ that represent neural networks:

$$w \quad \longmapsto \quad P(w).$$

- We study two sets of parameters on which AD can be incorrect:

$$\mathrm{inc}(P) = \left\{w \in \mathbb{M}^n : \mathcal{D}P(w) \text{ exists, but } \mathcal{D}^{\mathrm{AD}}P(w) \neq \mathcal{D}P(w)\right\}, \longleftarrow \text{incorrect set}$$

$$\mathrm{ndf}(P) = \{w \in \mathbb{M}^n : \mathcal{D}P(w) \text{ does not exist}\}. \longleftarrow \text{non-differentiable set}$$

Our Main Results

For any neural network P with ReLU activations and "bias parameters":

Theorem The incorrect set is always empty, i.e.,

$$|\mathrm{inc}(P)| = 0.$$

Our Main Results

For any neural network P with ReLU activations and "bias parameters":

Theorem The incorrect set is always empty, i.e.,

$$|\mathrm{inc}(P)| = 0.$$

Theorem The density of the non-differentiable set is bounded by

$$\frac{|\mathrm{ndf}(P)|}{|\mathbb{M}^n|} \leq \frac{(\#\ \text{ReLUs in } P)}{|\mathbb{M}|}.$$

This bound is tight up to a constant multiplicative factor.

Theorem On the non-differentiable set, AD computes a generalized derivative.

Our Main Results

piecewise analytic

For any neural network P with ~~ReLU~~ activations and “bias parameters”:

possibly without

Theorem The incorrect set is always empty, i.e.,

- We extend these results to more general neural networks.
  - Without bias parameters, these bounds become larger.
- We prove additional results such as:
  - Simple necessary & sufficient condition for deciding non-differentiability.

Theo

Moreover, this bound is tight up to a constant multiplicative factor.

For more details, read our paper and come to our poster session!

Theo