Brief overview of autodifferentiation

CS 690N, Spring 2017

Advanced Natural Language Processing

http://people.cs.umass.edu/~brenocon/anlp2017/

Brendan O'Connor

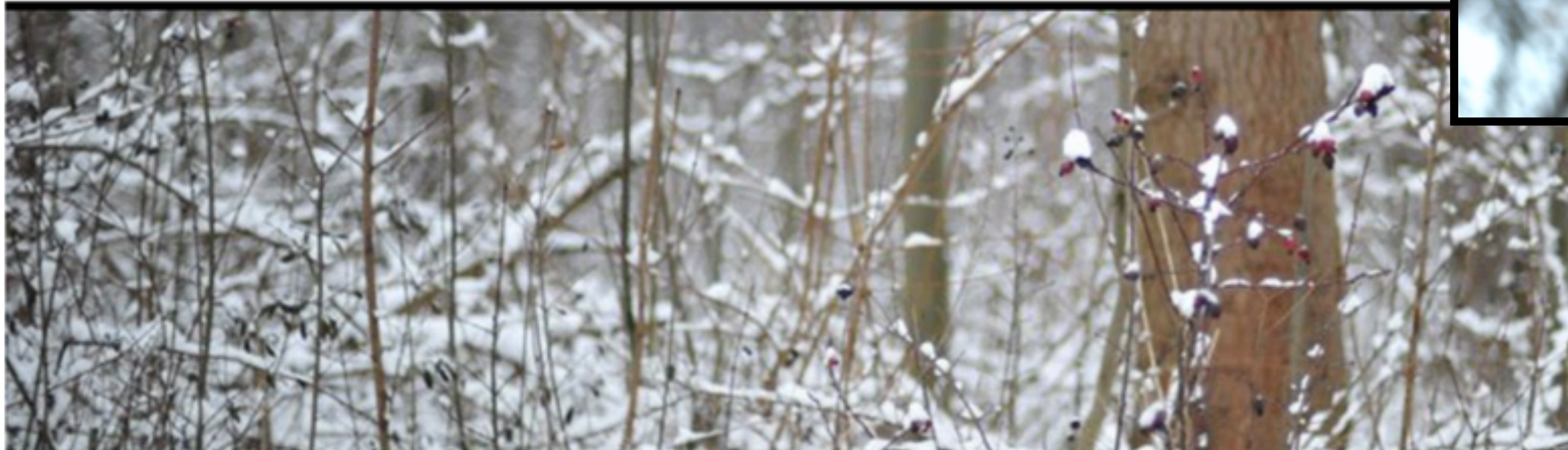
College of Information and Computer Sciences

University of Massachusetts Amherst

Automatic Differentiation: The ×

Secure https://justindomke.wordpress.com/2009/02/17/automatic-differentiation-t

Justin Domke's Weblog



Home About Backpropagation Completing the square in N dimensions Conditional Gradient Method Co

Automatic Differentiation: The most criminally underused tool in the potential machine learning toolbox?

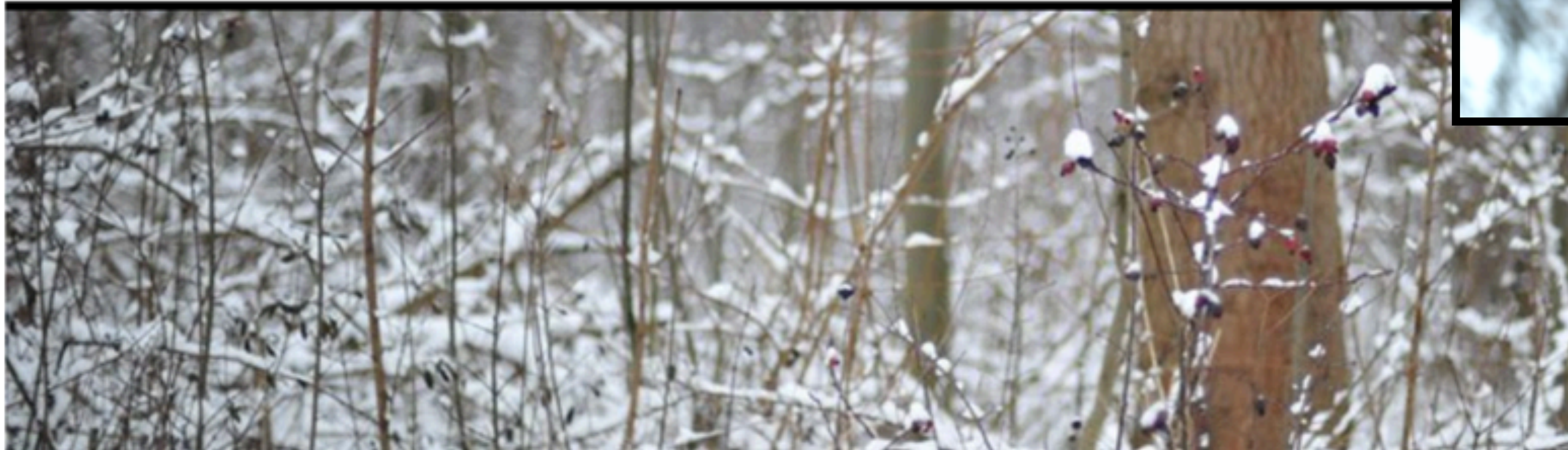
Posted on February 17, 2009

I recently got back reviews of a paper in which I used automatic differentiation. Therein, a reviewer clearly thought I was using finite difference, or "numerical" differentiation. This has led me to wondering: **Why don't machine learning people use automatic differentiation more? Why don't they use it…constantly?** Before recklessly speculating on the answer, let me briefly review what automatic differentiation

Automatic Differentiation: The ×

Secure https://justindomke.wordpress.com/2009/02/17/automatic-differentiation-t

Justin Domke's Weblog



Home About Backpropagation Completing the square in N dimensions Conditional Gradient Method Co

Automatic Differentiation: The most criminally underused tool in the potential machine learning toolbox?

Posted on February 17, 2009

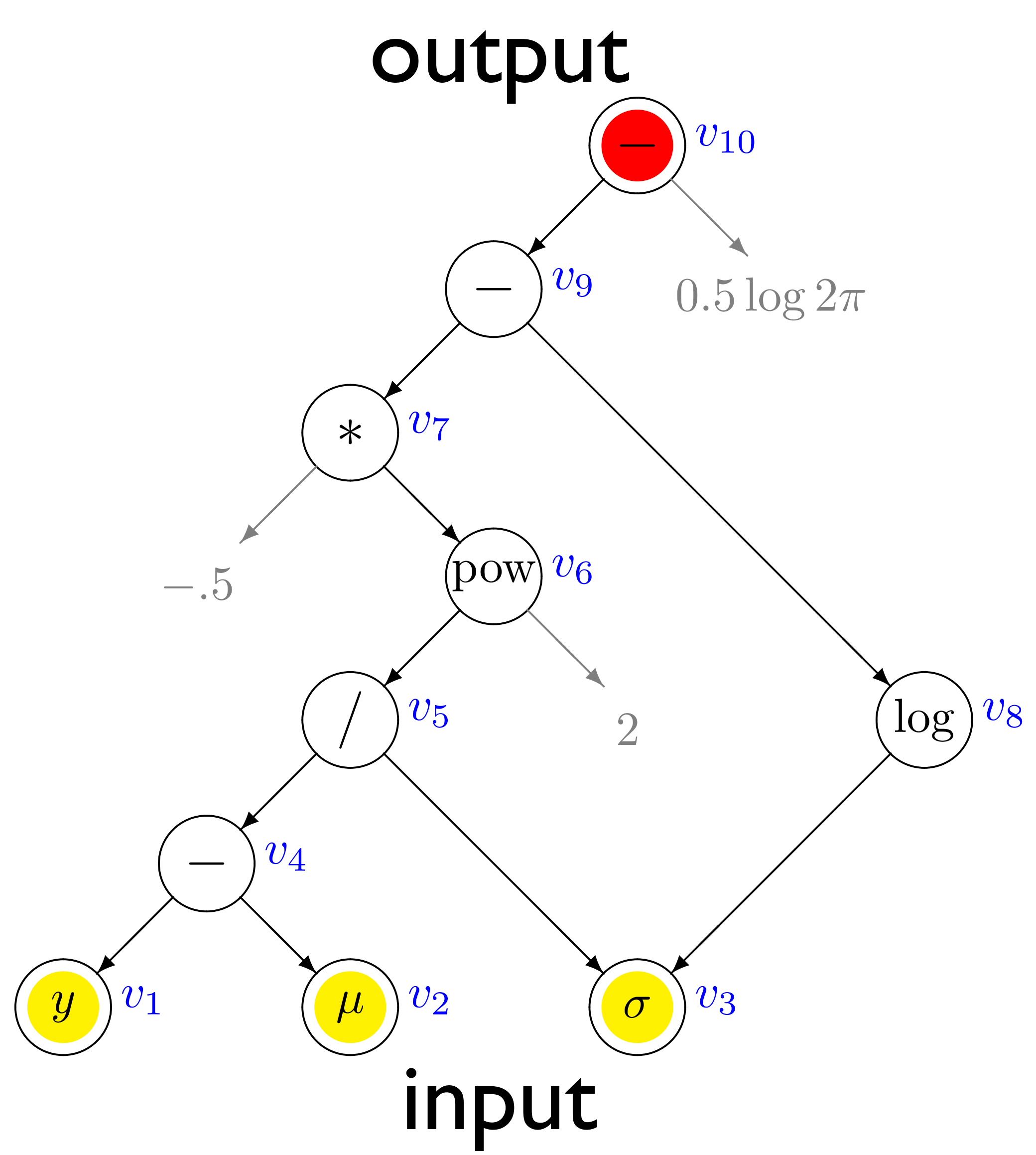
> *Update: (November 2015) In the almost seven years since writing this, there has been an explosion of great tools for automatic differentiation and a corresponding upsurge in its use. Thus, happily, this post is more or less obsolete.*

I recently got back reviews of a paper in which I used automatic differentiation. Therein, a reviewer clearly thought I was using finite difference, or "numerical" differentiation. This has led me to wondering: **Why don't machine learning people use automatic differentiation more? Why don't they use it…constantly?** Before recklessly speculating on the answer, let me briefly review what automatic differentiation

Goal: compute

$$\left(\frac{\partial f}{\partial x_1}(x_1,\ldots,x_N),\cdots,\frac{\partial f}{\partial x_N}(x_1,\ldots,x_N)\right)$$

$$f(y,\mu,\sigma) = -\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2 - \log\sigma - \frac{1}{2}\log(2\pi)$$



Forward mode

$$t_i = \sum_{j \in \text{children}[i]} \frac{\partial x_i}{\partial x_j} t_j$$

Reverse mode

$$a_j = \sum_{i \in \text{parents}[j]} \frac{\partial x_i}{\partial x_j} a_i$$

[Other strategies:
symbolic differentiation,
finite differences]

Forward mode

Table 2 Forward mode AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ at $(x_1, x_2) = (2, 5)$ and setting $\dot{x}_1 = 1$ to compute $\frac{\partial y}{\partial x_1}$. The original forward run on the left is augmented by the forward AD operations on the right, where each line supplements the original on its left.

Forward Evaluation Trace			Forward Derivative Trace		
v_{-1}	$= x_1$	$= 2$	$\dot{v}_{-1}$	$= \dot{x}_1$	$= 1$
v_0	$= x_2$	$= 5$	$\dot{v}_0$	$= \dot{x}_2$	$= 0$
v_1	$= \ln v_{-1}$	$= \ln 2$	$\dot{v}_1$	$= \dot{v}_{-1}/v_{-1}$	$= 1/2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 5$	$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + \dot{v}_0 \times v_{-1}$	$= 1 \times 5 + 0 \times 2$
v_3	$= \sin v_0$	$= \sin 5$	$\dot{v}_3$	$= \dot{v}_0 \times \cos v_0$	$= 0 \times \cos 5$
v_4	$= v_1 + v_2$	$= 0.693 + 10$	$\dot{v}_4$	$= \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
v_5	$= v_4 - v_3$	$= 10.693 + 0.959$	$\dot{v}_5$	$= \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
y	$= v_5$	$= 11.652$	$\boldsymbol{\dot{y}}$	$\boldsymbol{= \dot{v}_5}$	$\mathbf{= 5.5}$

Need to select which input you want derivative for

At each step calculate: **(v, v')**

Reverse mode

Table 3 Reverse mode AD example, with $y = f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$ at $(x_1, x_2) = (2, 5)$. After running the original forward run on the left, the augmented AD operations on the right are run in reverse (cf. Fig. 1). Both $\frac{\partial y}{\partial x_1}$ and $\frac{\partial y}{\partial x_2}$ are computed in the same reverse sweep, starting from the adjoint $\bar{v}_5 = \bar{y} = \frac{\partial y}{\partial y} = 1$.

Forward Evaluation Trace			Reverse Adjoint Trace			
$v_{-1} = x_1$		$= 2$	$\boldsymbol{\bar{x}_1 = \bar{v}_{-1}}$			$\mathbf{= 5.5}$
$v_0 = x_2$		$= 5$	$\boldsymbol{\bar{x}_2 = \bar{v}_0}$			$\mathbf{= 1.716}$
$v_1 = \ln v_{-1}$		$= \ln 2$	$\bar{v}_{-1} = \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \bar{v}_1 / v_{-1}$		$= 5.5$
			$\bar{v}_0 = \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$		$= 1.716$
$v_2 = v_{-1} \times v_0$		$= 2 \times 5$	$\bar{v}_{-1} = \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$		$= 5$
$v_3 = \sin v_0$		$= \sin 5$	$\bar{v}_0 = \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$		$= -0.284$
			$\bar{v}_2 = \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$		$= 1$
$v_4 = v_1 + v_2$		$= 0.693 + 10$	$\bar{v}_1 = \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$		$= 1$
$v_5 = v_4 - v_3$		$= 10.693 + 0.959$	$\bar{v}_3 = \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$		$= -1$
			$\bar{v}_4 = \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$		$= 1$
$y = v_5$		$= 11.652$	$\bar{v}_5 = \bar{y}$	$= 1$		

One sweep for gradient of f: R^n -> R

- All autodiff frameworks: write objective declaratively; gradients automatic
- Strategies
  - Statically compile a single computation graph (Theano, Tensorflow, Stan...)
    - Reapply graph to every datapoint
  - Dynamically make new graphs (DyNet, PyTorch...)
    - Different length sequences, parse trees, states within automaton data structures (e.g. shift-reduce) ...
- Uses
  - Stan: statistical modeling. MAP, Variational Bayes, MCMC
  - Currently in NLP, mostly just for NNs: MAP GD
    - Others possible?