THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

RELAXED SYSTEM LAB

Automatic Differentiation

COMP4901Y

Binhang Yuan

Numerical Differentiation

Numerical Differentiation

RELAXED SYSTEM LAB

- Numerical differentiation is the finite difference approximation of derivatives using values of the original function evaluated at some sample points.
- It is based on the limit definition of a derivative of function $f\colon \mathbb{R}^n \to \mathbb{R}$:

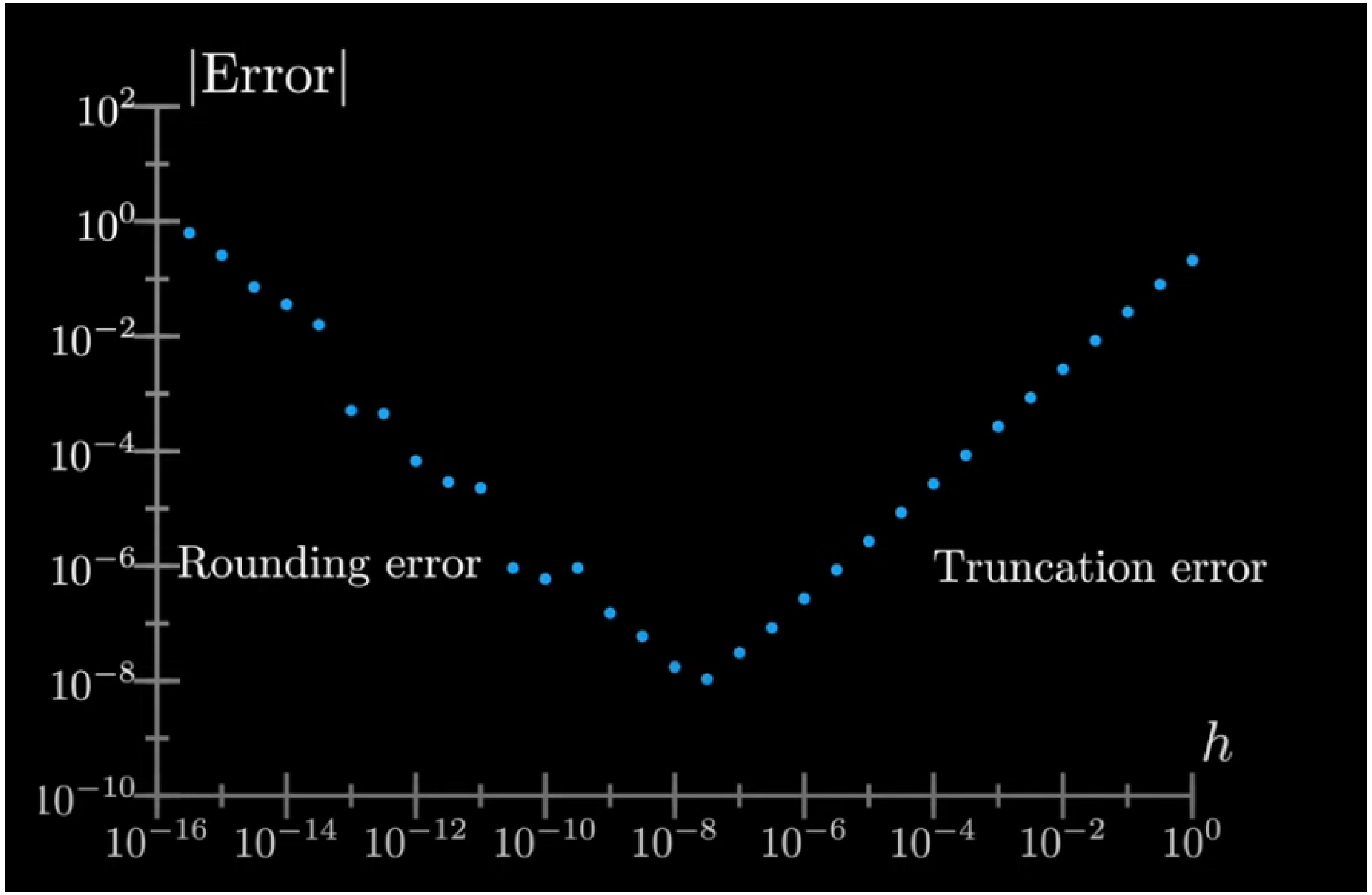
$$\frac{\partial f}{\partial x_i} = \lim_{\epsilon \to 0} \frac{f(\boldsymbol{x} + \epsilon \boldsymbol{e}_i) - f(\boldsymbol{x})}{\epsilon} \approx \frac{f(\boldsymbol{x} + h\boldsymbol{e}_i) - f(\boldsymbol{x})}{h}$$

- $\boldsymbol{e}_i$ is the i-th unit vector, $h > 0$ is a small step size.

Pros and Cons

- Advantage:
  - Easy to implement.
- Disadvantage:
  - Perform $\mathcal{O}(n)$ evaluatoins of f for a gradient in n dimensions.
  - Requires careful consideration in selecting the step size h.

Choose Step Size h



- Truncation Error:
  - The error of approximation that one gets from h not actually being zero.
  - Proportional to a power of h.
- Rounding Error:
  - The inaccuracy that is inflicted by the limited precision of computations.
  - Inversely proportional to a power of h.

Symbolic Differentiation

Derivative Computation Rules

- Derivative of sum or difference: $u = f(x), v = g(x)$:

$$\frac{d}{dx}(u \pm v) = \frac{du}{dx} \pm \frac{dv}{dx}$$

- Product Rule: $u = f(x), v = g(x)$:

$$\frac{d}{dx}(uv) = u\frac{dv}{dx} + v\frac{du}{dx}$$

- Chain Rule: $\mathrm{y} = f(u), u = g(x)$:

$$\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$$

Main Idea

RELAXED SYSTEM LAB

- Symbolic differentiation is the automatic manipulation of expressions for obtaining derivative expressions carried out by applying derivative computation rules.
- When formulae are represented as data structures, symbolically differentiating an expression tree is a perfectly mechanistic process.
- This is realized in modern computer algebra systems such as Mathematica.

Problem

RELAXED SYSTEM LAB

- Symbolic derivatives do not lend themselves to efficient runtime calculation of derivative values, as they can get exponentially larger than the expression whose derivative they represent.
- <u>Expression swell</u>: careless symbolic differentiation can easily produce exponentially large symbolic expressions that take correspondingly long to evaluate.

Expression Swell

Iterations of the logistic map $l_{n+1} = 4l_n(1 - l_n), l_1 = x$ and the corresponding derivatives of l_n with respect to x, illustrating expression swell.

n	l_n	$\frac{d}{dx}l_n$	$\frac{d}{dx}l_n$ (Simplified form)
1	x	1	1
2	$4x(1-x)$	$4(1-x)-4x$	$4-8x$
3	$16x(1-x)(1-2x)^2$	$16(1-x)(1-2x)^2 - 16x(1-2x)^2 - 64x(1-x)(1-2x)$	$16(1-10x+24x^2-16x^3)$
4	$64x(1-x)(1-2x)^2 (1-8x+8x^2)^2$	$128x(1-x)(-8+16x)(1-2x)^2(1-8x+8x^2)+64(1-x)(1-2x)^2(1-8x+8x^2)^2-64x(1-2x)^2(1-8x+8x^2)^2-256x(1-x)(1-2x)(1-8x+8x^2)^2$	$64(1-42x+504x^2-2640x^3+7040x^4-9984x^5+7168x^6-2048x^7)$

Automatic Differentiation

Main Idea

RELAXED SYSTEM LAB

- An automatic differentiation (AD) system will convert the program into a sequence of elementary operations with specified routines for computing derivatives:
  - Apply symbolic differentiation at the elementary operation level;
  - Keep intermediate numerical results;
  - Combining the derivatives of the constituent operations through the chain rule gives the derivative of the overall composition.

Notations

- The *Jacobian matrix* of a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ is defined by a $m \times n$ matrix noded by $\mathbf{J}$ where $\mathrm{J}_{ij} = \frac{\partial y_i}{\partial x_j}$, or explicitly:

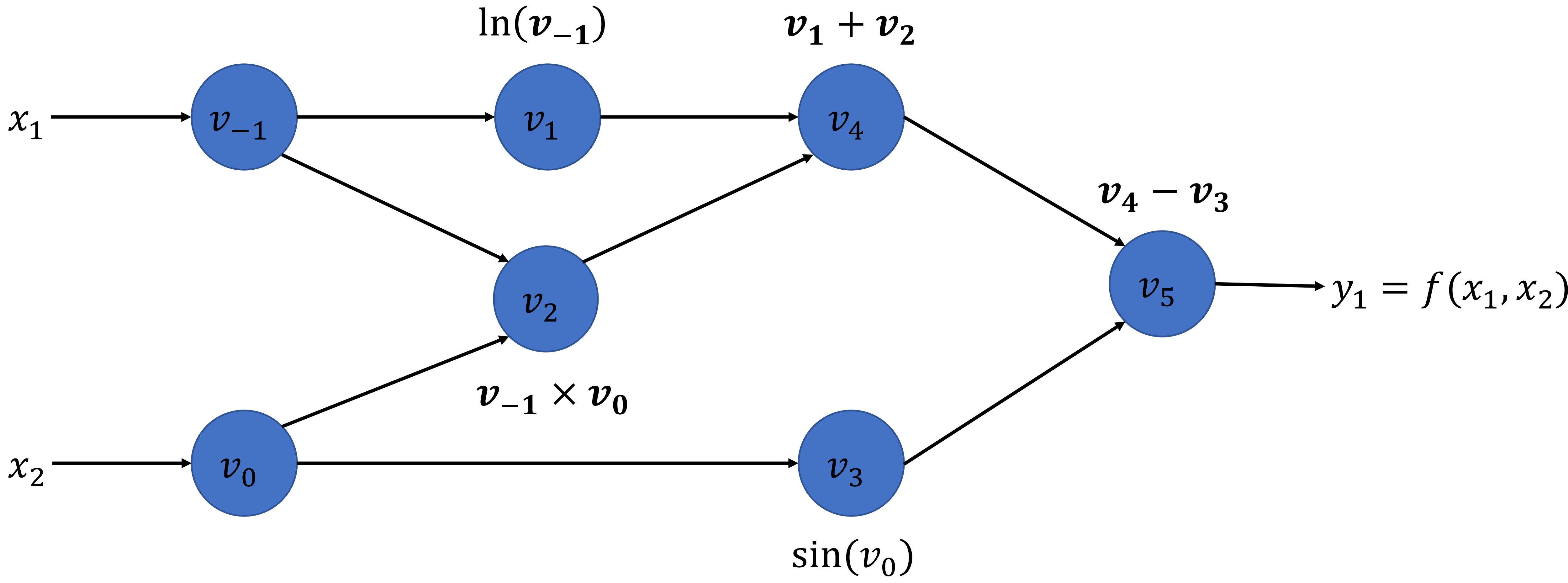
$$\mathbf{J} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$$

Notations

- A function $f\colon \mathbb{R}^n \rightarrow \mathbb{R}^m$ is constucted using intermidate variable v_i such that:
  - Variable $v_{j-n} = x_j,\ j = 1, \ldots, n$ are the input variables;
  - Variable $v_i,\ i = 1, \ldots, l$ are the intermidate variables;
  - Variable $y_{m-k} = v_{l-k},\ k = 1, \ldots, m$ are the output variables;

Example: $\boldsymbol{f}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \ln(\boldsymbol{x}_1) + \boldsymbol{x}_1\boldsymbol{x}_2 - \sin(\boldsymbol{x})$

RELAXED SYSTEM LAB



Forward Mode AD

RELAXED SYSTEM LAB

- For computing the derivative of f with respect to x_1, we start by associating with each intermediate variable v_i a derivative (tangent):

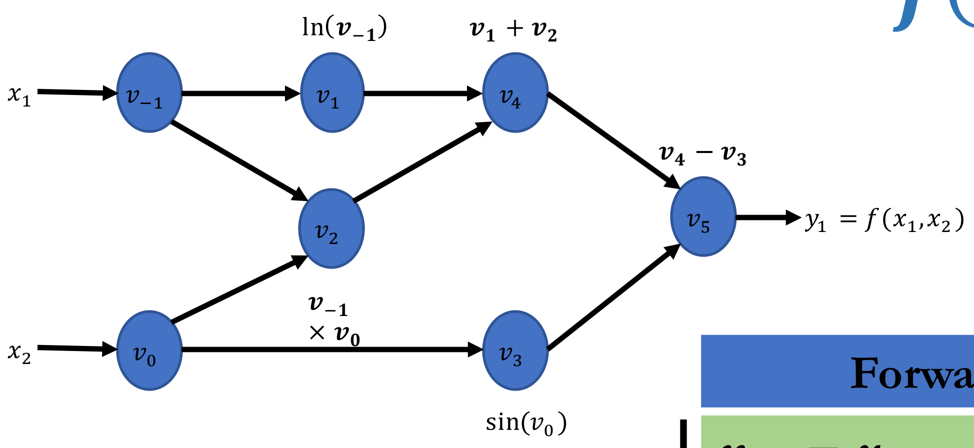
$$\dot{v}_i = \frac{\partial v_i}{\partial x_1}$$

- Apply the chain rule to each elementary operation in the forward primal trace;
- Generate the corresponding tangent (derivative) trace;
- Evaluating the primals v_i in lockstep with their corresponding tangents $\dot{v}_i$ gives us the required derivative in the final variable $\dot{v}_5 = \frac{\partial y_1}{\partial x_1}$.

Forward Mode AD:

RELAXED SYSTEM LAB

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x)$$



Forward Primal Trace	
$v_{-1} = x_1$	$= 2$
$v_0 = x_2$	$= 5$
$v_1 = \ln(x_2)$	$= \ln(5) = 0.693$
$v_2 = v_{-1} \times v_0$	$= 2 \times 5 = 10$
$v_3 = \sin v_0$	$= \sin 5 = 0.959$
$v_4 = v_1 + v_2$	$= 0.693 + 10$
$v_5 = v_4 - v_3$	$= 10.693 + 0.959$
$y_1 = v_5$	$= 11.652$

Forward Tangent (Derivative) Trace	
$\dot{v}_{-1} = \dot{x}_1$	$= 1$
$\dot{v}_0 = \dot{x}_2$	$= 0$
$\dot{v}_1 = \dot{v}_{-1}/v_{-1}$	$= 1/2$
$\dot{v}_2 = \dot{v}_{-1} \times v_0 + \dot{v}_0 \times v_{-1}$	$= 1 \times 5 + 0 \times 2$
$\dot{v}_3 = \dot{v}_0 \times \cos v_0$	$= 1 \times \cos 5$
$\dot{v}_4 = \dot{v}_1 + \dot{v}_2$	$= 0.5 + 5$
$\dot{v}_5 = \dot{v}_4 - \dot{v}_3$	$= 5.5 - 0$
$\dot{y}_1 = \dot{v}_5$	$= 5.5$

Forward Mode AD

- Compute the Jacobian of a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ with n independent/input variable x_i and m dependent/output variable y_j:
  - Each forward pass of AD is initialized by setting only one of the input variable x_i and setting the rest to 0 (i.e., $\dot{\boldsymbol{x}} = \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the i-th unit vector).
  - One exeucution of forward mode AD computes: $\dot{y}_j = \frac{\partial y_j}{\partial x_i}|_{\boldsymbol{x}=\boldsymbol{a}}, j = 1, \ldots, m$
  - Give us one columne of the Jacobian matrix at point $\boldsymbol{a}$ (the full jacobian can be computed by n evaluations):

$$\boldsymbol{J}_f = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}|_{\boldsymbol{x}=\boldsymbol{a}}$$

RELAXED SYSTEM LAB

Reverse Mode AD

- Reverse mode AD propagates derivatives backward from a given output.
- We start by complementing each intermediate variable v_i with an adjoint (cotangent) representing the sensitivity of a considered output y_j with respect to changes in v_i:

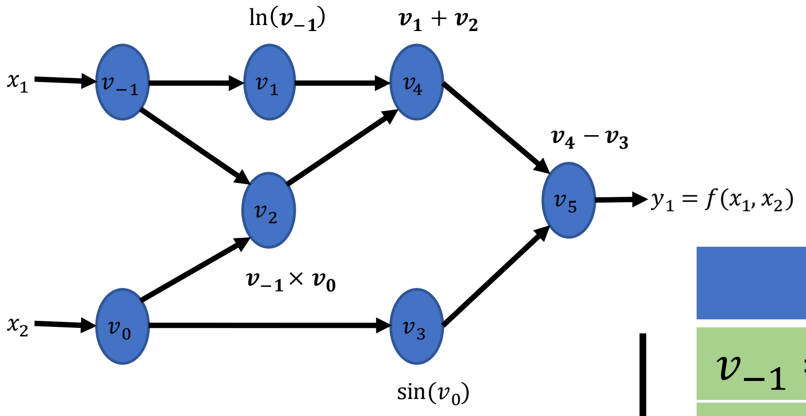
$$\bar{v}_i = \frac{\partial y_j}{\partial v_i}$$

- In the first phase, the original function code is run forward, populating intermediate variables v_i and recording the dependencies in the computational graph.
- In the second phase, derivatives are calculated by propagating adjoints $\bar{v}_i$ in reverse, from the outputs to the inputs.

Reverse Mode AD:

RELAXED SYSTEM LAB

$$f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x)$$



Forward Primal Trace	
$v_{-1} = x_1$	$= 2$
$v_0 = x_2$	$= 5$
$v_1 = \ln(x_2)$	$= \ln(5) = 0.693$
$v_2 = v_{-1} \times v_0$	$= 2 \times 5 = 10$
$v_3 = \sin v_0$	$= \sin 5 = 0.959$
$v_4 = v_1 + v_2$	$= 0.693 + 10$
$v_5 = v_4 - v_3$	$= 10.693 + 0.959$
$y_1 = v_5$	$= 11.652$

Reverse Adjoint (Derivative) Trace		
$\bar{x}_1 = \bar{v}_{-1}$		$= 5.5$
$\bar{x}_2 = \bar{v}_0$		$= 1.716$
$\bar{v}_{-1} = \bar{v}_{-1} + \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}}$	$= \bar{v}_{-1} + \bar{v}_1 / v_{-1}$	$= 5.5$
$\bar{v}_0 = \bar{v}_0 + \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_0 + \bar{v}_2 \times v_{-1}$	$= 1.716$
$\bar{v}_{-1} = \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_2 \times v_0$	$= 5$
$\bar{v}_0 = \bar{v}_3 \frac{\partial v_3}{\partial v_0}$	$= \bar{v}_3 \times \cos v_0$	$= -0.284$
$\bar{v}_2 = \bar{v}_4 \frac{\partial v_4}{\partial v_2}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_1 = \bar{v}_4 \frac{\partial v_4}{\partial v_1}$	$= \bar{v}_4 \times 1$	$= 1$
$\bar{v}_3 = \bar{v}_5 \frac{\partial v_5}{\partial v_3}$	$= \bar{v}_5 \times (-1)$	$= -1$
$\bar{v}_4 = \bar{v}_5 \frac{\partial v_5}{\partial v_4}$	$= \bar{v}_5 \times 1$	$= 1$
$\bar{v}_5 = \bar{y}_1$		$= 1$

Reverse Mode AD

- Compute the Jacobian of a function $f\colon \mathbb{R}^n \to \mathbb{R}^m$ with n independent/input variable x_i and m dependent/output variable y_j.
- An important advantage of the reverse mode is that it is significantly less costly to evaluate (in terms of operation count) than the forward mode for functions with a large number of inputs.
- In the extreme case of $f\colon \mathbb{R}^n \to \mathbb{R}$ only one application of the reverse mode is sufficient to compute the full gradient.
- Because machine learning practice principally involves the gradient of a scalar-valued objective with respect to a large number of parameters, this establishes the reverse mode as the main technique in ML systems.

References

Journal of Machine Learning Research 18 (2018) 1-43 Submitted 8/17; Published 4/18

Automatic Differentiation in Machine Learning: a Survey

Atılım Güneş Baydin GUNES@ROBOTS.OX.AC.UK
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ, United Kingdom

Barak A. Pearlmutter BARAK@PEARLMUTTER.NET
Department of Computer Science
National University of Ireland Maynooth
Maynooth, Co. Kildare, Ireland

Alexey Andreyevich Radul AXCH@MIT.EDU
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139, United States

Jeffrey Mark Siskind QOBI@PURDUE.EDU
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907, United States

- Automatic differentiation in machine learning: a survey (https://arxiv.org/abs/1502.05767)