Introduction to Automatic Differentiation

Luca Manzoni

Differentiation: possible techniques

- By hand
- Numerical
- Symbolic
- Automatic Differentiation

Differentiation: by hand

- The derivative is computed “offline”, the result is then coded
- As done with the original backpropagation
- You do not want to do it

Differentiation: numerical

- You can approximate the derivative $\frac{\partial f}{\partial x_i}$ with $\frac{f(x + h\mathbf{e}_i) - f(x)}{h}$ for a small value of h

- Pros: easy to implement

Differentiation: numerical

Cons: numerical instability

Sum of a small number to a possibly large one

Subtraction of two numbers of similar magnitude

$$\frac{f(x + h\mathbf{e}_i) - f(x)}{h}$$

Division by a number near zero

Some techniques allow to reduce the approximation error (but are far from perfect)

Differentiation: numerical

Cons: computational cost

Let $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^m$. We can compute the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Each derivative requires 2 evaluations of the function...

...for a total of 2mn evaluations

Differentiation: symbolic

- You can use a symbolic differentiation engine to compute exactly the derivative
- Available in multiple libraries and CAS (e.g., Mathematica, SimPy, ...)
- Pros: no approximation!

Differentiation: symbolic

- Cons: difficult to manage selection (if) and loops (for, while)
- Cons: the symbolic representation of the derivative can grow too large!

Automatic Differentiation

- A way to obtain the exact value of the derivative at a certain point
- The computation is augmented by keeping track some additional values for all intermediate steps of the computation

Automatic Differentiation

- Two (main) ways of performing automatic differentiation:
    - Forward mode
      (AKA Tangent Linear Mode)
    - Reverse mode
      (AKA Adjoint or Cotangent Linear Mode)

A (first) Running Example

We will use a function $g\colon \mathbb{R} \to \mathbb{R}$ defined as follows:

$$g(x) = \cos(5x^2)$$

But, since we usually have multiple inputs and outputs…

A (second) Running Example

We will use a function $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ defined as follows:

$$f(x_1, x_2) = (f_1(x_1, x_2), f_2(x_1, x_2))$$

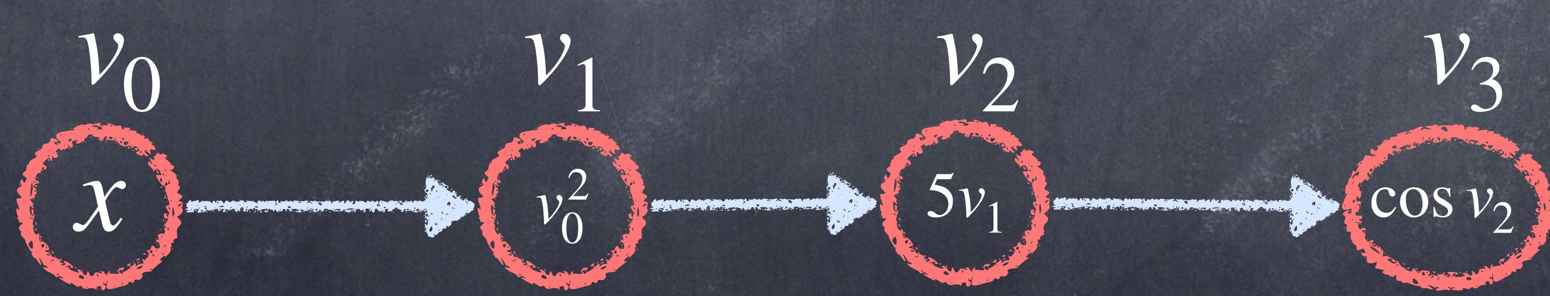
With:

$$f_1(x_1, x_2) = x_1 x_2 + \cos x_1$$

$$f_2(x_1, x_2) = x_2^3 + \ln x_1 - x_2$$

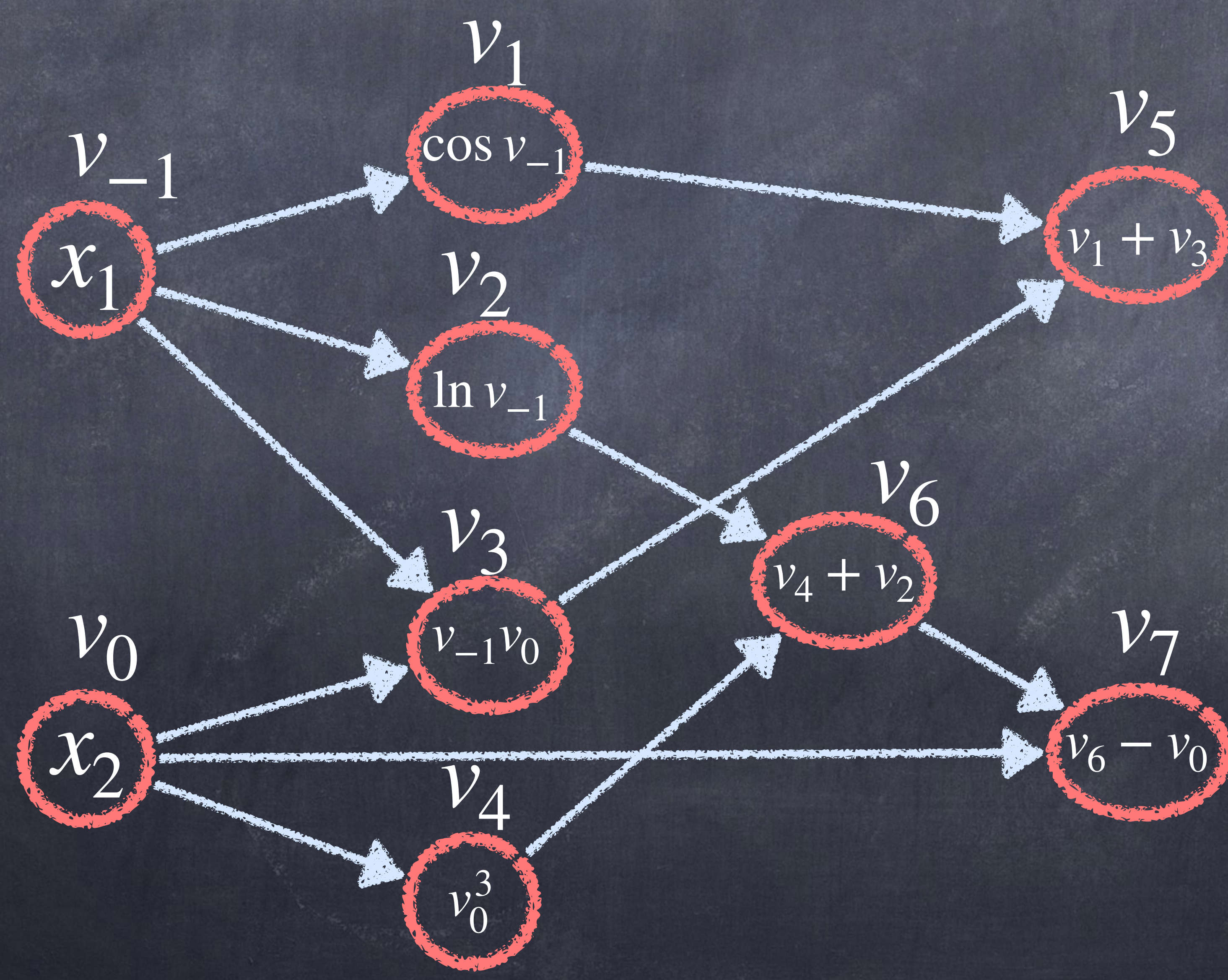
Computational graph

We can represent the function g with a graph where every intermediate operation is assigned to a variable



Computational graph

The same can be done with f:



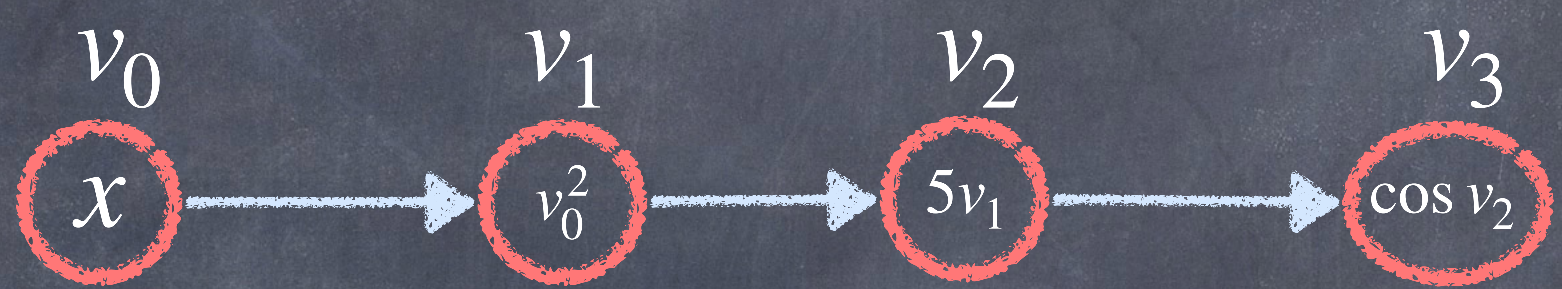
Forward-Mode AutoDiff

- The information “moves” from the inputs to the outputs
- Suppose that we want to derive w.r.t. the input x_j
- Then, each variable v_i has an associated value $\dot{v}_i$ which is $\dfrac{\partial v_i}{\partial x_j}$

Forward-Mode AutoDiff

- We compute all v_i, keeping track of the values
  (obtaining the forward primal trace)
- We can compute all $\dot{v}_i$ using only the values in the primal trace and the already computed $\dot{v}_k$ for $k < i$

Forward mode



Forward Primal Trace

$$v_0 = 2$$
$$v_1 = v_0^2 = 2^2 = 4$$
$$v_2 = 5v_1 = 5 \times 4 = 20$$
$$v_3 = \cos v_2 = \cos 20 = 0.408$$

Forward Tangent Trace

$$\dot{v}_0 = 1$$
$$\dot{v}_1 = \frac{\partial v_1}{\partial v_0} = 2v_0 = 4$$
$$\dot{v}_2 = \frac{\partial v_2}{\partial v_0} = \frac{\partial v_2}{\partial v_1}\frac{\partial v_1}{\partial v_0} = \frac{\partial v_2}{\partial v_1}\dot{v}_1 = 5\dot{v}_0 = 20$$
$$\dot{v}_3 = \frac{\partial v_3}{\partial v_0} = \frac{\partial v_3}{\partial v_2}\frac{\partial v_2}{\partial v_0} = \frac{\partial v_3}{\partial v_2}\dot{v}_2 = -\sin(v_2)\dot{v}_2 = -18.259$$

Forward mode

Forward Primal Trace

$v_{-1} = 2$

$v_0 = 3$

$v_1 = \cos v_{-1} = -0.416$

$v_2 = \ln v_{-1} = 0.693$

$v_3 = v_{-1}v_0 = 6$

$v_4 = v_0^3 = 27$

$v_5 = v_1 + v_3 = 5.584$

$v_6 = v_4 + v_2 = 27.693$

$v_7 = v_6 - v_0 = 24.693$

The two outputs (y_1 and y_2)

Forward Tangent Trace

Now we must decide if we want to differentiate w.r.t x_1 or x_2 (we select x_1)

$\dot{v}_{-1} = 1$

$\dot{v}_0 = 0$

$$\dot{v}_1 = \frac{\partial v_1}{\partial v_{-1}}\dot{v}_{-1} = -\sin(v_{-1})\dot{v}_{-1} = -0.909$$

$$\dot{v}_2 = \frac{\partial v_2}{\partial v_{-1}}\dot{v}_{-1} = \frac{1}{v_{-1}}\dot{v}_{-1} = 0.5$$

$$\dot{v}_3 = \frac{\partial v_2}{\partial v_{-1}}\dot{v}_{-1} + \frac{\partial v_2}{\partial v_0}\dot{v}_0 = v_0\dot{v}_{-1} = 3$$

$$\dot{v}_4 = \frac{\partial v_4}{\partial v_0}\dot{v}_0 = 0$$

$$\dot{v}_5 = \frac{\partial v_5}{\partial v_1}\dot{v}_1 + \frac{\partial v_5}{\partial v_3}\dot{v}_3 = \dot{v}_1 + \dot{v}_3 = 2.090$$

$$\dot{v}_6 = \frac{\partial v_6}{\partial v_4}\dot{v}_4 + \frac{\partial v_6}{\partial v_2}\dot{v}_2 = \dot{v}_2 = 0.5$$

$$\dot{v}_7 = \frac{\partial v_7}{\partial v_6}\dot{v}_6 + \frac{\partial v_7}{\partial v_0}\dot{v}_0 = \dot{v}_6 = 0.5$$

The derivatives $\frac{\partial y_1}{\partial x_1}$ and $\frac{\partial y_2}{\partial x_1}$

Forward-Mode: things to notice

- By setting $\dot{x}_i = 1$ and $\dot{x}_j = 0$ for all $j \neq i$ we can compute the derivative of all outputs w.r.t. x_i
- To compute w.r.t. each input variable we must repeat the process multiple times

Forward-Mode: things to notice

- All derivative are of simple "basic" operations (sums, products, trigonometric functions)
- We can compute any composition of them via the forward-mode diff
- The value obtained is the exact value of the derivative*

*There can still be floating point approximations, but they are of a different kind w.r.t. the one obtained when computing the derivative numerically

Forward-Mode: things to notice

- There is no obstacle in performing the derivation with loops and conditionals
- For the forward mode we can actually compute the derivatives at the same time as the computation of the forward primal trace

Forward mode: Jacobian

Let $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^m$. We can compute the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \boxed{\begin{matrix} \dfrac{\partial f_1}{\partial x_1} \\ \dfrac{\partial f_2}{\partial x_1} \\ \vdots \\ \dfrac{\partial f_m}{\partial x_1} \end{matrix}} & \begin{matrix} \dfrac{\partial f_1}{\partial x_2} \\ \dfrac{\partial f_2}{\partial x_2} \\ \\ \dfrac{\partial f_m}{\partial x_2} \end{matrix} & \begin{matrix} \cdots \\ \cdots \\ \\ \cdots \end{matrix} & \begin{matrix} \dfrac{\partial f_1}{\partial x_n} \\ \dfrac{\partial f_2}{\partial x_n} \\ \vdots \\ \dfrac{\partial f_m}{\partial x_n} \end{matrix} \end{bmatrix}$$

Each "pass" allow us to compute a column of the Jacobian matrix...

...using a a total of n evaluations

Which is good when n is small w.r.t. m

Forward mode: Jacobian-vector product

Let $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^m$ and let $\mathbf{r} \in \mathbb{R}^n$. We can compute the product $\mathbf{J}\,\mathbf{r}$ without computing the Jacobian matrix

$$\mathbf{J}\mathbf{r} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \longrightarrow$$

Start the computation of the Forward Tangent Trace with $\dot{x}_1 = r_1, \dot{x}_2 = r_2, \ldots$ i.e., $\dot{\mathbf{x}} = \mathbf{r}$

Dual Numbers

The forward-mode differentiation can be interpreted as working with an extension of the real numbers, called dual numbers

Dual numbers are of the form: $v + \dot{v}\epsilon$

Where $\epsilon \neq 0$ but $\epsilon^2 = 0$

Notice that addition and multiplication works as expected:

$$(v + \dot{v}\epsilon) + (u + \dot{u}\epsilon) = (v + u) + (\dot{v} + \dot{u})\epsilon$$

$$\begin{aligned}(v + \dot{v}\epsilon)(u + \dot{u}\epsilon) &= vu + v\dot{u}\epsilon + \dot{v}u\epsilon + \dot{v}\dot{u}\epsilon^2 \\ &= vu + (v\dot{u} + \dot{v}u)\epsilon\end{aligned}$$

Dual Numbers

Suppose that for each function f the following holds:

$$f(v + \dot{v}\epsilon) = f(v) + f'(v)\dot{v}\epsilon$$

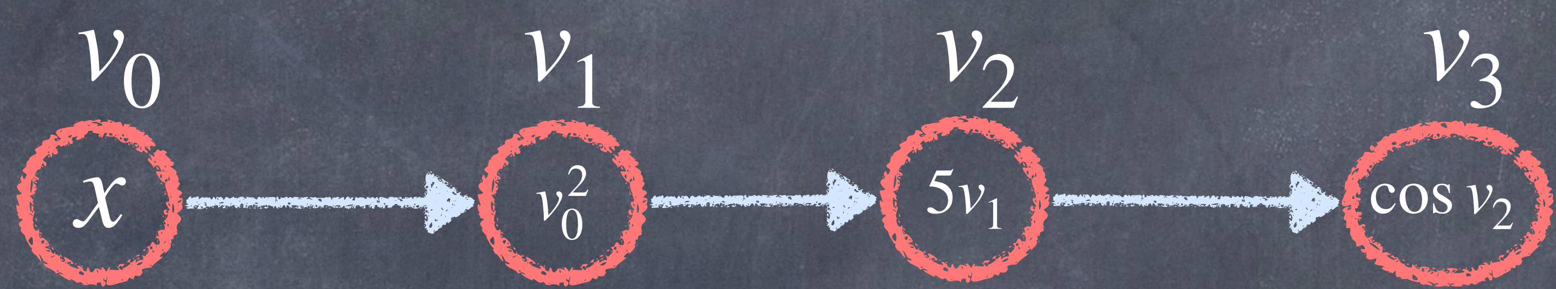
Then two applications of the previous property give us the chain rule:

$$\begin{aligned} f(g(v + \dot{v}\epsilon)) &= f(g(v) + g'(v)\dot{v}\epsilon) \\ &= f(g(v)) + f'(g(v))g'(v)\dot{v}\epsilon \end{aligned}$$

Reverse-Mode AutoDiff

- Fix one of the outputs y_j
- In reverse-mode we add to each variable the adjoint $\bar{v}_i = \dfrac{\partial y_j}{\partial v_i}$
- Notice that this time we change the variable w.r.t. the derivative is computed instead of keeping it fixed

Reverse mode



Forward Primal Trace

$$v_0 = 2$$
$$v_1 = v_0^2 = 2^2 = 4$$
$$v_2 = 5v_1 = 5 \times 4 = 20$$
$$v_3 = \cos v_2 = \cos 20 = 0.408$$

Reverse Adjoint Trace

$$\bar{v}_3 = 1$$
$$\bar{v}_2 = \frac{\partial y}{\partial v_2} = \frac{\partial y}{\partial v_3}\frac{\partial v_3}{\partial v_2} = -\bar{v}_3 - \sin(v_2) = -0.913$$
$$\bar{v}_1 = \frac{\partial y}{\partial v_1} = \frac{\partial y}{\partial v_2}\frac{\partial v_2}{\partial v_1} = \bar{v}_2\frac{\partial v_2}{\partial v_1} = 5\bar{v}_2 = -4.565$$
$$\bar{v}_0 = \frac{\partial y}{\partial v_0} = \frac{\partial y}{\partial v_1}\frac{\partial v_1}{\partial v_0} = \bar{v}_1\frac{\partial v_1}{\partial v_0} = 2v_0\bar{v}_1 = -18.259$$

Reverse mode

Forward Primal Trace

$v_{-1} = 2$

$v_0 = 3$

$v_1 = \cos v_{-1} = -0.416$

$v_2 = \ln v_{-1} = 0.693$

$v_3 = v_{-1} v_0 = 6$

$v_4 = v_0^3 = 27$

$v_5 = v_1 + v_3 = 5.584$

$v_6 = v_4 + v_2 = 27.693$

$v_7 = v_6 - v_0 = 24.693$

The two outputs
(y_1 and y_2)

Reverse Adjoint Trace

Now we must decide
the output that
we want to differentiate
(we select y_1)

$\bar{v}_5 = 1$

$\bar{v}_7 = 0$

$$\bar{v}_6 = \frac{\partial y_1}{\partial v_6} = \frac{\partial y_1}{\partial v_7}\frac{\partial v_7}{\partial v_6} = \bar{v}_7 \frac{\partial v_7}{\partial v_6} = 0$$

$$\bar{v}_4 = \frac{\partial y_1}{\partial v_4} = \frac{\partial y_1}{\partial v_6}\frac{\partial v_6}{\partial v_4} = \bar{v}_6 \frac{\partial v_6}{\partial v_4} = 0$$

$$\bar{v}_3 = \frac{\partial y_1}{\partial v_3} = \frac{\partial y_1}{\partial v_5}\frac{\partial v_5}{\partial v_3} = \bar{v}_5 \frac{\partial v_5}{\partial v_3} = 1$$

$$\bar{v}_2 = \frac{\partial y_1}{\partial v_2} = \frac{\partial y_1}{\partial v_6}\frac{\partial v_6}{\partial v_2} = \bar{v}_6 \frac{\partial v_6}{\partial v_2} = 0$$

$$\bar{v}_1 = \frac{\partial y_1}{\partial v_1} = \frac{\partial y_1}{\partial v_5}\frac{\partial v_5}{\partial v_1} = \bar{v}_5 \frac{\partial v_5}{\partial v_1} = 1$$

$$\bar{v}_0 = \frac{\partial y_1}{\partial v_0} = \frac{\partial y_1}{\partial v_3}\frac{\partial v_3}{\partial v_0} + \frac{\partial y_1}{\partial v_4}\frac{\partial v_4}{\partial v_0} = \bar{v}_3 \frac{\partial v_3}{\partial v_0} + \bar{v}_4 \frac{\partial v_4}{\partial v_0} = \bar{v}_3 v_{-1} = 2$$

$$\bar{v}_{-1} = \frac{\partial y_1}{\partial v_{-1}} = \frac{\partial y_1}{\partial v_1}\frac{\partial v_1}{\partial v_{-1}} + \frac{\partial y_1}{\partial v_2}\frac{\partial v_2}{\partial v_{-1}} + \frac{\partial y_1}{\partial v_3}\frac{\partial v_3}{\partial v_{-1}}$$

$$= \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}} + \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}} + \bar{v}_3 \frac{\partial v_3}{\partial v_{-1}} = -\sin(v_{-1}) + v_0 = 2.090$$

The derivatives
$\frac{\partial y_1}{\partial x_2}$ and $\frac{\partial y_1}{\partial x_1}$

Reverse-Mode: things to notice

- By setting $\bar{y}_i = 1$ and $\bar{y}_j = 0$ for all $j \neq i$ we can compute the derivatives of the output y_i w.r.t. all inputs
- To compute w.r.t. each output variable we must repeat the process multiple times

Reverse-Mode: things to notice

- The other observations done for forward-mode autodiff also holds for the reverse-mode autodiff
- You might have noticed that the procedure used is a generalisation of the one employed by backpropagation

Reverse mode: Jacobian

Let $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^m$. We can compute the Jacobian matrix:

$$\mathbf{J} = \begin{bmatrix} \boxed{\frac{\partial f_1}{\partial x_1} \quad \frac{\partial f_1}{\partial x_2} \quad \cdots \quad \frac{\partial f_1}{\partial x_n}} \\ \frac{\partial f_2}{\partial x_1} \quad \frac{\partial f_2}{\partial x_2} \quad \cdots \quad \frac{\partial f_2}{\partial x_n} \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ \frac{\partial f_m}{\partial x_1} \quad \frac{\partial f_m}{\partial x_2} \quad \cdots \quad \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Each "pass" allow us to compute a row of the Jacobian matrix...

...using a a total of m evaluations

Which is good when m is small w.r.t. n

Reverse mode: transposed Jacobian-vector product

Let $\mathbf{f}\colon \mathbb{R}^n \to \mathbb{R}^m$ and let $\mathbf{r} \in \mathbb{R}^m$. We can compute the product $\mathbf{J}^T\mathbf{r}$ without computing the transpose of the Jacobian matrix

$$\mathbf{J}^T\mathbf{r} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_2} \\ \vdots & & & \vdots \\ \frac{\partial f_1}{\partial x_n} & \frac{\partial f_2}{\partial x_n} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}$$

Start the computation of the Reverse Adjoint Trace with $\bar{y}_1 = r_1, \bar{y}_2 = r_2, \ldots$ i.e., $\bar{\mathbf{y}} = \mathbf{r}$