ECE49595NL Lecture 34: Automatic Differentiation—III

Jeffrey Mark Siskind

Elmore Family School of Electrical and Computer Engineering

Spring 2025

PURDUE UNIVERSITY | Elmore Family School of Electrical and Computer Engineering

A Neural Network

$x_0 \quad \frac{\partial x_8}{\partial x_0}$

layer$_0$ θ_0 w_0

$x_1 \quad \frac{\partial x_8}{\partial x_1}$

layer$_1$ θ_1 w_1

$x_2 \quad \frac{\partial x_8}{\partial x_2}$

layer$_2$ θ_2 w_2

$x_3 \quad \frac{\partial x_8}{\partial x_3}$

layer$_3$ θ_3 w_3

$x_4 \quad \frac{\partial x_8}{\partial x_4}$

layer$_4$ θ_4 w_4

$x_5 \quad \frac{\partial x_8}{\partial x_5}$

layer$_5$ θ_5 w_5

$x_6 \quad \frac{\partial x_8}{\partial x_6}$

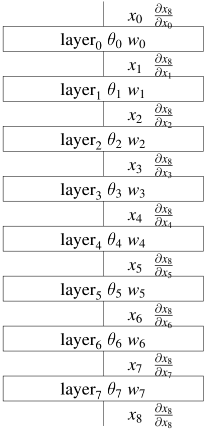
layer$_6$ θ_6 w_6

$x_7 \quad \frac{\partial x_8}{\partial x_7}$

layer$_7$ θ_7 w_7

$x_8 \quad \frac{\partial x_8}{\partial x_8}$

A Neural Network is a (Functional) Program



$$
\begin{array}{l}
\text{net}\ [\theta_0,\ldots,\theta_7]\ [w_0,\ldots,w_7]\ x_0 \triangleq\\
\quad \textbf{let}\ \begin{array}{rcl}
x_1 & = & \text{layer}_0\ \theta_0\ w_0\ x_0\\
x_2 & = & \text{layer}_1\ \theta_1\ w_1\ x_1\\
x_3 & = & \text{layer}_2\ \theta_2\ w_2\ x_2\\
x_4 & = & \text{layer}_3\ \theta_3\ w_3\ x_3\\
x_5 & = & \text{layer}_4\ \theta_4\ w_4\ x_4\\
x_6 & = & \text{layer}_5\ \theta_5\ w_5\ x_5\\
x_7 & = & \text{layer}_6\ \theta_6\ w_6\ x_6\\
x_8 & = & \text{layer}_7\ \theta_7\ w_7\ x_7
\end{array}\\
\quad \textbf{in}\ x_8
\end{array}
$$

A (Functional) Program

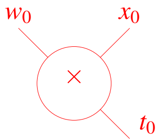
$$
\begin{array}{l}
f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
\qquad \textbf{let}\ \begin{array}{rcl} t_0 & = & w_0 \times x_0 \\ t_1 & = & w_1 \times x_1 \\ y & = & t_0 + t_1 \end{array} \\
\qquad \textbf{in}\ y
\end{array}
$$

A (Functional) Program is a (Neural) Network

$$
\begin{array}{l}
f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
\qquad \textbf{let}\ \begin{array}{rcl} t_0 & = & w_0 \times x_0 \\ t_1 & = & w_1 \times x_1 \\ y & = & t_0 + t_1 \end{array} \\
\qquad \textbf{in}\ y
\end{array}
$$

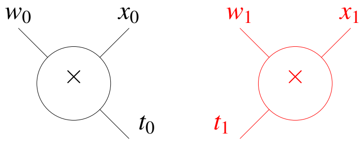
A (Functional) Program is a (Neural) Network

$$
\begin{aligned}
&f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
&\quad \textbf{let}\ \ t_0 = w_0 \times x_0 \\
&\quad\phantom{\textbf{let}}\ \ t_1 = w_1 \times x_1 \\
&\quad\phantom{\textbf{let}}\ \ \ y = t_0 + t_1 \\
&\quad \textbf{in}\ y
\end{aligned}
$$



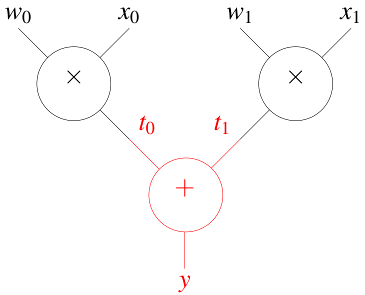
A (Functional) Program is a (Neural) Network

$$
\begin{aligned}
&f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
&\quad \textbf{let}\ \ t_0 = w_0 \times x_0 \\
&\quad \phantom{\textbf{let}\ \ } t_1 = w_1 \times x_1 \\
&\quad \phantom{\textbf{let}\ \ } y = t_0 + t_1 \\
&\quad \textbf{in}\ y
\end{aligned}
$$



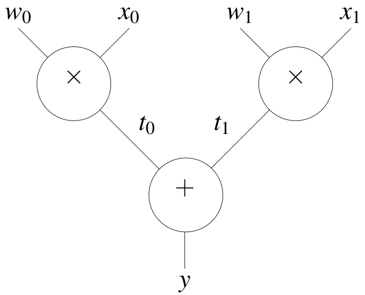
A (Functional) Program is a (Neural) Network

$$
\begin{aligned}
&f\ [w_0, w_1]\ [x_0, x_1] \triangleq\\
&\quad \textbf{let}\ \ t_0 = w_0 \times x_0\\
&\quad\phantom{\textbf{let}}\ \ t_1 = w_1 \times x_1\\
&\quad\phantom{\textbf{let}}\ \ {\color{red} y = t_0 + t_1}\\
&\quad \textbf{in}\ y
\end{aligned}
$$

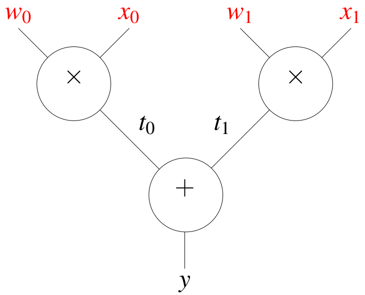


A (Functional) Program is a (Neural) Network

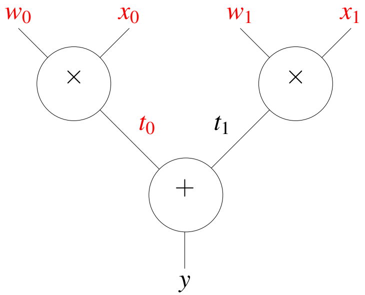
$$
\begin{array}{l}
f\ [w_0, w_1]\ [x_0, x_1] \triangleq \\
\quad \textbf{let}\ \ \begin{array}{rcl} t_0 & = & w_0 \times x_0 \\ t_1 & = & w_1 \times x_1 \\ y & = & t_0 + t_1 \end{array} \\
\quad \textbf{in}\ y
\end{array}
$$



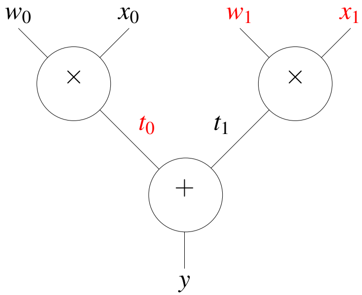
Evaluating a Network



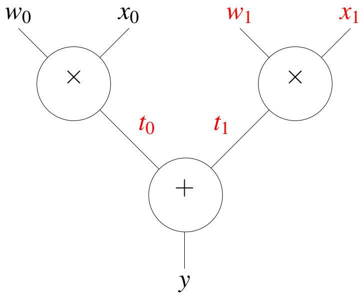
Evaluating a Network



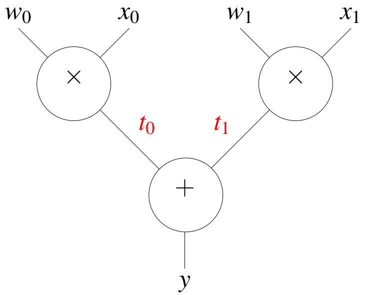
Evaluating a Network



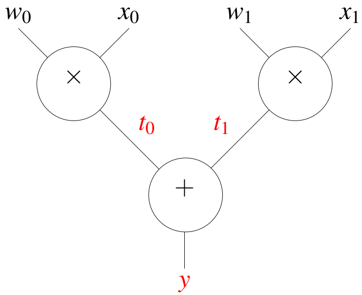
Evaluating a Network



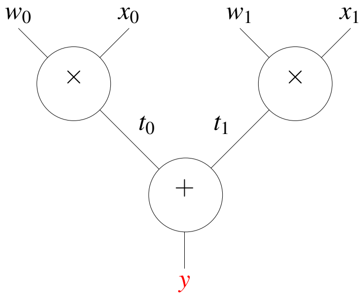
Evaluating a Network



Evaluating a Network



Evaluating a Network



Matrix multiplication and transposition

$$(\mathbf{X}_1 \times \mathbf{X}_2)^\top = \mathbf{X}_2^\top \times \mathbf{X}_1^\top$$

Programs as function composition

$$f = f_1 \circ \cdots \circ f_n$$

$$\begin{aligned}
\mathbf{x}_1 &= f_1\ \mathbf{x}_0 \\
&\vdots \\
\mathbf{x}_n &= f_n\ \mathbf{x}_{n-1}
\end{aligned}$$

The chain rule applied to programs

$$\mathcal{J}\ f\ \mathbf{x}_0 = (\mathcal{J}\ f_n\ \mathbf{x}_{n-1}) \times \cdots \times (\mathcal{J}\ f_1\ \mathbf{x}_0)$$
$$(\mathcal{J}\ f\ \mathbf{x}_0)^\top = (\mathcal{J}\ f_1\ \mathbf{x}_0)^\top \times \cdots \times (\mathcal{J}\ f_n\ \mathbf{x}_{n-1})^\top$$

Computing the Jacobian

$$
\begin{aligned}
\overrightharpoon{\mathbf{X}_1} &= (\mathcal{J}\, f_1\ \mathbf{x}_0) \\
\overrightharpoon{\mathbf{X}_2} &= (\mathcal{J}\, f_2\ \mathbf{x}_1) \times \overrightharpoon{\mathbf{X}_1} \\
&\ \vdots \\
\overrightharpoon{\mathbf{X}_n} &= (\mathcal{J}\, f_n\ \mathbf{x}_{n-1}) \times \overrightharpoon{\mathbf{X}_{n-1}}
\end{aligned}
$$

Computing the transpose of the Jacobian

$$
\begin{aligned}
\grave{\mathbf{X}}_{n-1} &= (\mathcal{J}\, f_n\ \mathbf{x}_{n-1})^\top \\
\grave{\mathbf{X}}_{n-2} &= (\mathcal{J}\, f_{n-1}\ \mathbf{x}_{n-2})^\top \times \grave{\mathbf{X}}_{n-1} \\
&\vdots \\
\grave{\mathbf{X}}_{0} &= (\mathcal{J}\, f_1\ \mathbf{x}_{0})^\top \times \grave{\mathbf{X}}_{1}
\end{aligned}
$$

Unary machine-state transition functions

$$\mathbf{x}[L_i] := u_i\ \mathbf{x}[R_i]$$

Binary machine-state transition functions

$$\mathbf{x}[L_i] := b\left(\mathbf{x}[R_i], \mathbf{x}[S_i]\right)$$

What a unary machine-state transition function does

$$\mathbf{x}_{i-1} = (\quad \boxed{\mathbf{x}_{i-1}[1]} \cdots \boxed{\mathbf{x}_{i-1}[L_i]} \cdots \boxed{\mathbf{x}_{i-1}[R_i]} \cdots \boxed{\mathbf{x}_{i-1}[m]} \,)$$

$$\mathbf{x}_i = f_i\ \mathbf{x}_{i-1} = (\, \boxed{\mathbf{x}_{i-1}[1]} \cdots \boxed{u_i\ \mathbf{x}_{i-1}[R_i]} \cdots \boxed{\mathbf{x}_{i-1}[R_i]} \cdots \boxed{\mathbf{x}_{i-1}[m]} \,)$$

What a binary machine-state transition function does

$$\mathbf{x}_{i-1} = (\quad \boxed{\mathbf{x}_{i-1}[1]} \cdots \quad \boxed{\mathbf{x}_{i-1}[L_i]} \quad \cdots \boxed{\mathbf{x}_{i-1}[R_i]} \cdots \boxed{\mathbf{x}_{i-1}[S_i]} \cdots \boxed{\mathbf{x}_{i-1}[m]})$$

$$\mathbf{x}_i = f_i\ \mathbf{x}_{i-1} = (\boxed{\mathbf{x}_{i-1}[1]} \cdots \boxed{b_i\ (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i])} \cdots \boxed{\mathbf{x}_{i-1}[R_i]} \cdots \boxed{\mathbf{x}_{i-1}[S_i]} \cdots \boxed{\mathbf{x}_{i-1}[m]})$$

Computing a Jacobian-vector product

$$\overrightharpoon{\mathbf{x}_n} = (\mathcal{J}\, f\; \mathbf{x}_0) \times \overrightharpoon{\mathbf{x}_0}$$

$$\begin{aligned}
\overrightharpoon{\mathbf{x}_1} &= (\mathcal{J}\, f_1\; \mathbf{x}_0) \times \overrightharpoon{\mathbf{x}_0} \\
&\vdots \\
\overrightharpoon{\mathbf{x}_n} &= (\mathcal{J}\, f_n\; \mathbf{x}_{n-1}) \times \overrightharpoon{\mathbf{x}_{n-1}}
\end{aligned}$$

Computing a vector-Jacobian product

$$\overline{\mathbf{x}_0} = (\mathcal{J}\; f\; \mathbf{x}_0)^\top \times \overline{\mathbf{x}_n}$$

$$\begin{aligned}
\overline{\mathbf{x}_{n-1}} &= (\mathcal{J}\; f_n\; \mathbf{x}_{n-1})^\top \times \overline{\mathbf{x}_n}\\
&\vdots\\
\overline{\mathbf{x}_0} &= (\mathcal{J}\; f_1\; \mathbf{x}_0)^\top \times \overline{\mathbf{x}_1}
\end{aligned}$$

Jacobian of a unary primitive

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} & & & L_i & & & R_i & & \\ & & & \downarrow & & & \downarrow & & \end{array} \\
\begin{array}{c} \\ \\ \\ L_i \rightarrow \\ \\ \\ R_i \rightarrow \\ \\ \\ \end{array} &
\left[\begin{array}{ccccccccc}
1 & & & & & & & & \\
 & \ddots & & & & & & & \\
 & & 1 & & & & & & \\
 & & & 0 & & & u' & & \\
 & & & & 1 & & & & \\
 & & & & & \ddots & & & \\
 & & & & & & 1 & & \\
 & & & & & & & \ddots & \\
 & & & & & & & & 1
\end{array}\right]
\end{array}
$$

$$u' = \mathcal{D}\ u_i\ \mathbf{x}_{i-1}[R_i]$$

Jacobian of a binary primitive

$$
\begin{array}{c|ccccccccccc|}
 & & & & L_i & & & R_i & & S_i & & \\
 & & & & \downarrow & & & \downarrow & & \downarrow & & \\
\hline
 & 1 & & & & & & & & & & \\
 & & \ddots & & & & & & & & & \\
 & & & 1 & & & & & & & & \\
L_i \rightarrow & & & & 0 & & & b'_1 & & b'_2 & & \\
 & & & & & 1 & & & & & & \\
 & & & & & & \ddots & & & & & \\
R_i \rightarrow & & & & & & & 1 & & & & \\
 & & & & & & & & \ddots & & & \\
S_i \rightarrow & & & & & & & & & 1 & & \\
 & & & & & & & & & & \ddots & \\
 & & & & & & & & & & & 1 \\
\hline
\end{array}
$$

$$
\begin{aligned}
b'_1 &= \mathcal{D}_1\ b_i\ (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i]) \\
b'_2 &= \mathcal{D}_2\ b_i\ (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i])
\end{aligned}
$$

Unary single-step Jacobian-vector product

$$
\begin{pmatrix}
\overrightarrow{\mathbf{x}_{i-1}}[1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i - 1] \\
u' \times \overrightarrow{\mathbf{x}_{i-1}}[R_i] \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i + 1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[R_i] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[m]
\end{pmatrix}
=
\begin{pmatrix}
1 & & & & & & & & \\
& \ddots & & & & & & & \\
& & 1 & & & & & & \\
& & & 0 & & & u' & & \\
& & & & 1 & & & & \\
& & & & & \ddots & & & \\
& & & & & & 1 & & \\
& & & & & & & \ddots & \\
& & & & & & & & 1
\end{pmatrix}
\begin{pmatrix}
\overrightarrow{\mathbf{x}_{i-1}}[1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i - 1] \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i] \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i + 1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[R_i] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[m]
\end{pmatrix}
$$

$$u' = \mathcal{D}\; u_i\; \mathbf{x}_{i-1}[R_i]$$

Binary single-step Jacobian-vector product

$$
\begin{pmatrix}
\overrightarrow{\mathbf{x}_{i-1}}[1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i - 1] \\
b'_1 \times \overrightarrow{\mathbf{x}_{i-1}}[R_i] + b'_2 \times \overrightarrow{\mathbf{x}_{i-1}}[S_i] \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i + 1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[R_i] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[S_i] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[m]
\end{pmatrix}
=
\begin{pmatrix}
1 & & & & & & & & & & \\
 & \ddots & & & & & & & & & \\
 & & 1 & & & & & & & & \\
 & & & 0 & & & b'_1 & & b'_2 & & \\
 & & & & 1 & & & & & & \\
 & & & & & \ddots & & & & & \\
 & & & & & & 1 & & & & \\
 & & & & & & & \ddots & & & \\
 & & & & & & & & 1 & & \\
 & & & & & & & & & \ddots & \\
 & & & & & & & & & & 1
\end{pmatrix}
\begin{pmatrix}
\overrightarrow{\mathbf{x}_{i-1}}[1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i - 1] \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i] \\
\overrightarrow{\mathbf{x}_{i-1}}[L_i + 1] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[R_i] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[S_i] \\
\vdots \\
\overrightarrow{\mathbf{x}_{i-1}}[m]
\end{pmatrix}
$$

$$
\begin{aligned}
b'_1 &= \mathcal{D}_1\ b_i\ (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i]) \\
b'_2 &= \mathcal{D}_2\ b_i\ (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i])
\end{aligned}
$$

Unary single-step vector-Jacobian product

$$\begin{pmatrix} \overleftharpoon{\mathbf{x}}_i[1] \\ \vdots \\ \overleftharpoon{\mathbf{x}}_i[L_i-1] \\ 0 \\ \overleftharpoon{\mathbf{x}}_i[L_i+1] \\ \vdots \\ u' \times \overleftharpoon{\mathbf{x}}_i[L_i] + \overleftharpoon{\mathbf{x}}_i[R_i] \\ \vdots \\ \overleftharpoon{\mathbf{x}}_i[m] \end{pmatrix} = \begin{pmatrix} 1 &&&&&&&& \\ & \ddots &&&&&&& \\ && 1 &&&&&& \\ &&& 0 &&&&& \\ &&&& 1 &&&& \\ &&&&& \ddots &&& \\ &&& u' &&& 1 && \\ &&&&&&& \ddots & \\ &&&&&&&& 1 \end{pmatrix} \begin{pmatrix} \overleftharpoon{\mathbf{x}}_i[1] \\ \vdots \\ \overleftharpoon{\mathbf{x}}_i[L_i-1] \\ \overleftharpoon{\mathbf{x}}_i[L_i] \\ \overleftharpoon{\mathbf{x}}_i[L_i+1] \\ \vdots \\ \overleftharpoon{\mathbf{x}}_i[R_i] \\ \vdots \\ \overleftharpoon{\mathbf{x}}_i[m] \end{pmatrix}$$

$$u' = \mathcal{D}\; u_i\; \mathbf{x}_{i-1}[R_i]$$

Binary single-step vector-Jacobian product

$$
\begin{pmatrix}
\overleftarrow{\mathbf{x}}_i[1] \\
\vdots \\
\overleftarrow{\mathbf{x}}_i[L_i-1] \\
0 \\
\overleftarrow{\mathbf{x}}_i[L_i+1] \\
\vdots \\
b'_1 \times \overleftarrow{\mathbf{x}}_i[L_i] + \overleftarrow{\mathbf{x}}_i[R_i] \\
\vdots \\
b'_2 \times \overleftarrow{\mathbf{x}}_i[L_i] + \overleftarrow{\mathbf{x}}_i[S_i] \\
\vdots \\
\overleftarrow{\mathbf{x}}_i[m]
\end{pmatrix}
=
\begin{pmatrix}
1 &&&&&&&&&& \\
& \ddots &&&&&&&&& \\
&& 1 &&&&&&&& \\
&&& 0 &&&&&&& \\
&&&& 1 &&&&&& \\
&&&&& \ddots &&&&& \\
&&& b'_1 &&& 1 &&&& \\
&&&&&&& \ddots &&& \\
&&& b'_2 &&&&& 1 && \\
&&&&&&&&& \ddots & \\
&&&&&&&&&& 1
\end{pmatrix}
\begin{pmatrix}
\overleftarrow{\mathbf{x}}_i[1] \\
\vdots \\
\overleftarrow{\mathbf{x}}_i[L_i-1] \\
\overleftarrow{\mathbf{x}}_i[L_i] \\
\overleftarrow{\mathbf{x}}_i[L_i+1] \\
\vdots \\
\overleftarrow{\mathbf{x}}_i[R_i] \\
\vdots \\
\overleftarrow{\mathbf{x}}_i[S_i] \\
\vdots \\
\overleftarrow{\mathbf{x}}_i[m]
\end{pmatrix}
$$

$$
\begin{aligned}
b'_1 &= \mathcal{D}_1\; b_i\; (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i]) \\
b'_2 &= \mathcal{D}_2\; b_i\; (\mathbf{x}_{i-1}[R_i], \mathbf{x}_{i-1}[S_i])
\end{aligned}
$$

Forward Mode

$$f = f_1 \circ \cdots \circ f_n$$

Forward Mode

$$f = f_1 \circ \cdots \circ f_n$$
$$\mathcal{J}(f)(x_0) = \mathcal{J}(f_n)(x_{n-1}) \times \cdots \times \mathcal{J}(f_1)(x_0)$$

Forward Mode

$$
\begin{aligned}
f &= f_1 \circ \cdots \circ f_n \\
\mathcal{J}(f)(x_0) &= \mathcal{J}(f_n)(x_{n-1}) \times \cdots \times \mathcal{J}(f_1)(x_0) \\
\acute{x}_n &= \mathcal{J}(f)(x_0) \times \acute{x}_0
\end{aligned}
$$

Forward Mode

$$\begin{aligned}
f &= f_1 \circ \cdots \circ f_n \\
\mathcal{J}(f)(x_0) &= \mathcal{J}(f_n)(x_{n-1}) \times \cdots \times \mathcal{J}(f_1)(x_0) \\
\acute{x}_n &= \mathcal{J}(f)(x_0) \times \acute{x}_0
\end{aligned}$$

$$\begin{aligned}
x_1 &= f_1(x_0) \\
\acute{x}_1 &= \mathcal{J}(f_1)(x_0) \times \acute{x}_0 \\
&\vdots \\
x_n &= f_n(x_{n-1}) \\
\acute{x}_n &= \mathcal{J}(f_n)(x_{n-1}) \times \acute{x}_{n-1}
\end{aligned}$$

Reverse Mode

$$f = f_1 \circ \cdots \circ f_n$$

Reverse Mode

$$
\begin{aligned}
f &= f_1 \circ \cdots \circ f_n \\
\mathcal{J}(f)(x_0)^\top &= \mathcal{J}(f_1)(x_0)^\top \times \cdots \times \mathcal{J}(f_n)(x_{n-1})^\top
\end{aligned}
$$

Reverse Mode

$$f = f_1 \circ \cdots \circ f_n$$
$$\mathcal{J}(f)(x_0)^\top = \mathcal{J}(f_1)(x_0)^\top \times \cdots \times \mathcal{J}(f_n)(x_{n-1})^\top$$
$$\grave{x}_0 = \mathcal{J}(f)(x_0)^\top \times \grave{x}_n$$

Reverse Mode

$$
\begin{aligned}
f &= f_1 \circ \cdots \circ f_n \\
\mathcal{J}(f)(x_0)^\top &= \mathcal{J}(f_1)(x_0)^\top \times \cdots \times \mathcal{J}(f_n)(x_{n-1})^\top \\
\grave{x}_0 &= \mathcal{J}(f)(x_0)^\top \times \grave{x}_n
\end{aligned}
$$

$$
\begin{aligned}
x_1 &= f_1(x_0) \\
&\vdots \\
x_n &= f_n(x_{n-1}) \\
\grave{x}_{n-1} &= \mathcal{J}(f_n)(x_{n-1}) \times \grave{x}_n \\
&\vdots \\
\grave{x}_0 &= \mathcal{J}(f_1)(x_0) \times \grave{x}_1
\end{aligned}
$$

Forward Mode by Overloading

$$
\begin{aligned}
x_1 &= f_1(x_0)\\
\acute{x}_1 &= \mathcal{J}(f_1)(x_0) \times \acute{x}_0\\
&\vdots\\
x_n &= f_n(x_{n-1})\\
\acute{x}_n &= \mathcal{J}(f_n)(x_{n-1}) \times \acute{x}_{n-1}
\end{aligned}
$$

Forward Mode by Overloading

$$
\begin{aligned}
x_1 &= f_1(x_0)\\
\acute{x}_1 &= \mathcal{J}(f_1)(x_0) \times \acute{x}_0\\
&\vdots\\
x_n &= f_n(x_{n-1})\\
\acute{x}_n &= \mathcal{J}(f_n)(x_{n-1}) \times \acute{x}_{n-1}
\end{aligned}
$$

$$x_i = f_i(x_{i-1})$$

Forward Mode by Overloading

$$\begin{aligned}
x_1 &= f_1(x_0)\\
\acute{x}_1 &= \mathcal{J}(f_1)(x_0) \times \acute{x}_0\\
&\vdots\\
x_n &= f_n(x_{n-1})\\
\acute{x}_n &= \mathcal{J}(f_n)(x_{n-1}) \times \acute{x}_{n-1}
\end{aligned}$$

$$\begin{aligned}
x_i &= f_i(x_{i-1})\\
\langle x_i, \acute{x}_i\rangle &= \langle f_i(x_{i-1}), \mathcal{J}(f_i)(x_{i-1}) \times \acute{x}_{i-1}\rangle
\end{aligned}$$

Forward Mode by Overloading

$$
\begin{aligned}
x_1 &= f_1(x_0)\\
\acute{x}_1 &= \mathcal{J}(f_1)(x_0) \times \acute{x}_0\\
&\vdots\\
x_n &= f_n(x_{n-1})\\
\acute{x}_n &= \mathcal{J}(f_n)(x_{n-1}) \times \acute{x}_{n-1}
\end{aligned}
$$

$$
\begin{aligned}
x_i &= f_i(x_{i-1})\\
\langle x_i, \acute{x}_i\rangle &= \langle f_i(x_{i-1}), \mathcal{J}(f_i)(x_{i-1}) \times \acute{x}_{i-1}\rangle\\
\overrightarrow{x_i} &= \overrightarrow{f_i}\,(\overrightarrow{x_{i-1}})
\end{aligned}
$$

Reverse Mode

$$
\begin{aligned}
x_1 &= f_1(x_0)\\
&\vdots\\
x_n &= f_n(x_{n-1})\\
\grave{x}_{n-1} &= \mathcal{J}(f_n)(x_{n-1}) \times \grave{x}_n\\
&\vdots\\
\grave{x}_0 &= \mathcal{J}(f_1)(x_0) \times \grave{x}_1
\end{aligned}
$$