CS 4803 / 7643: Deep Learning

Topics:

- Automatic Differentiation
  - (Finish) Forward mode vs Reverse mode AD
  - Patterns in backprop

Dhruv Batra

Georgia Tech

Administrativia

- HW1 Reminder
  - Due: 09/09, 11:59pm

- HW2 out on 9/10
  - Schedule: https://www.cc.gatech.edu/classes/AY2021/cs7643_fall/

- Project discussion next class

Recap from last time

How do we compute gradients?

- Analytic or “Manual” Differentiation
- ✗ Symbolic Differentiation
- ✓ Numerical Differentiation
- Automatic Differentiation
  - Forward mode AD
  - Reverse mode AD
    - aka “backprop”

Vector/Matrix Derivatives Notation

$x, y \in \mathbb{R}$

$\vec{x} \in \mathbb{R}^d \quad \vec{y} \in \mathbb{R}^c$

$X, Y \in \mathbb{R}^{m \times n}$

$$\begin{array}{c|ccc} y \backslash x & S & V & M \\ \hline S & \frac{\partial y}{\partial x} & \frac{\partial y}{\partial \vec{x}} & \frac{\partial y}{\partial X} \\ V & \frac{\partial \vec{y}}{\partial x} & \frac{\partial \vec{y}}{\partial \vec{x}} & \text{tensors} \\ M & \frac{\partial Y}{\partial x} & & \end{array}$$

$$\frac{\partial \vec{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \vdots \\ \frac{\partial y_c}{\partial x} \end{bmatrix}_{c \times 1}$$

num = $\underbrace{\text{dim } 1}_{\text{rows}}$

den = $\underbrace{\text{dim } 2}_{\text{col}}$

$$\boxed{\frac{\partial L}{\partial w}} \quad \boxed{\frac{\partial y}{\partial \vec{x}}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} & \cdots & \frac{\partial y}{\partial x_d} \end{bmatrix}$$

Vector/Matrix Derivatives Notation

$$\frac{\partial \vec{y}}{\partial \vec{x}} = \begin{array}{cc} & \begin{array}{ccccc} & & j & & \end{array} \\ i & \begin{bmatrix} \frac{\partial y_i}{\partial x_1} & \cdots & \frac{\partial y_i}{\partial x_j} & \cdots & \frac{\partial y_i}{\partial x_d} \\ & & \frac{\partial y_c}{\partial x_j} & & \end{bmatrix}_{c \times d} \end{array}$$

$$\boxed{\text{Jacobian}}$$

Vector Derivative Example

$$\frac{\partial(\vec{w}^T A \vec{w})}{\partial \vec{w}} = 2\vec{w}^T \boxed{A}$$

$$\vec{y} = A\vec{x}$$

$$\boxed{\frac{\partial \vec{y}}{\partial \vec{x}}} = \boxed{A}$$

$$y_i = \sum_j a_{ij} x_j$$

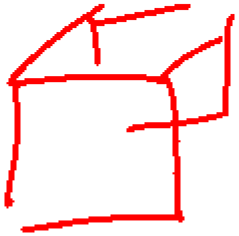
$$i \cdot \left[\frac{\partial y_i}{\partial x_j} \right] = \left[\cdot\cdot\ a_{ij} \right]$$

$$\frac{\partial y_i}{\partial x_j} = a_{ij}$$

Extension to Tensors

$\underline{X} \in \underline{\mathbb{R}^{d_1 \times \dots \times d_n}}$

$\underline{Y} \in \underline{\mathbb{R}^{c_1 \times \dots \times c_n}}$



$\to$ y-vec = $\boxed{Y(:)}$

$\to$ x-vec = $\boxed{X(:)}$

$$\frac{\partial Y[i_1, \dots, i_m]}{\partial X[j_1, \dots, j_m]}$$

$$\frac{\partial \overrightarrow{y\text{-vec}}}{\partial \overrightarrow{x\text{-vec}}} = \begin{bmatrix} & (\,,\,) & \\ & (\,) \partial & \\ (\,+) & & \end{bmatrix}$$

"Jacobian Matrix"

Chain Rule: Composite Functions

$$f(g(x)) = (f \circ g)(x)$$

$\leftarrow$ layers in NN

$$\overbrace{f(x)}^{L(\vec{w})} = g_l(g_{l-1} \cdots g_1(x))$$

$$= (g_l \circ g_{l-1} \circ \cdots g_1)(x)$$

Chain Rule: Scalar Case

$$\underline{x} \in \mathbb{R}^1 \xrightarrow[g_1(\,)]{} \underline{z} \in \mathbb{R}^1 \xrightarrow[g_2(\,)]{} \underline{y} \in \mathbb{R}^1$$

$$y = g_2(\underbrace{g_1(x)}_{z})$$

$$\frac{\partial y}{\partial \underline{x}} = \left(\frac{\partial y}{\partial \underline{z}}\right) \cdot \left(\frac{\partial z}{\partial \underline{x}}\right)$$

Scalar mult.

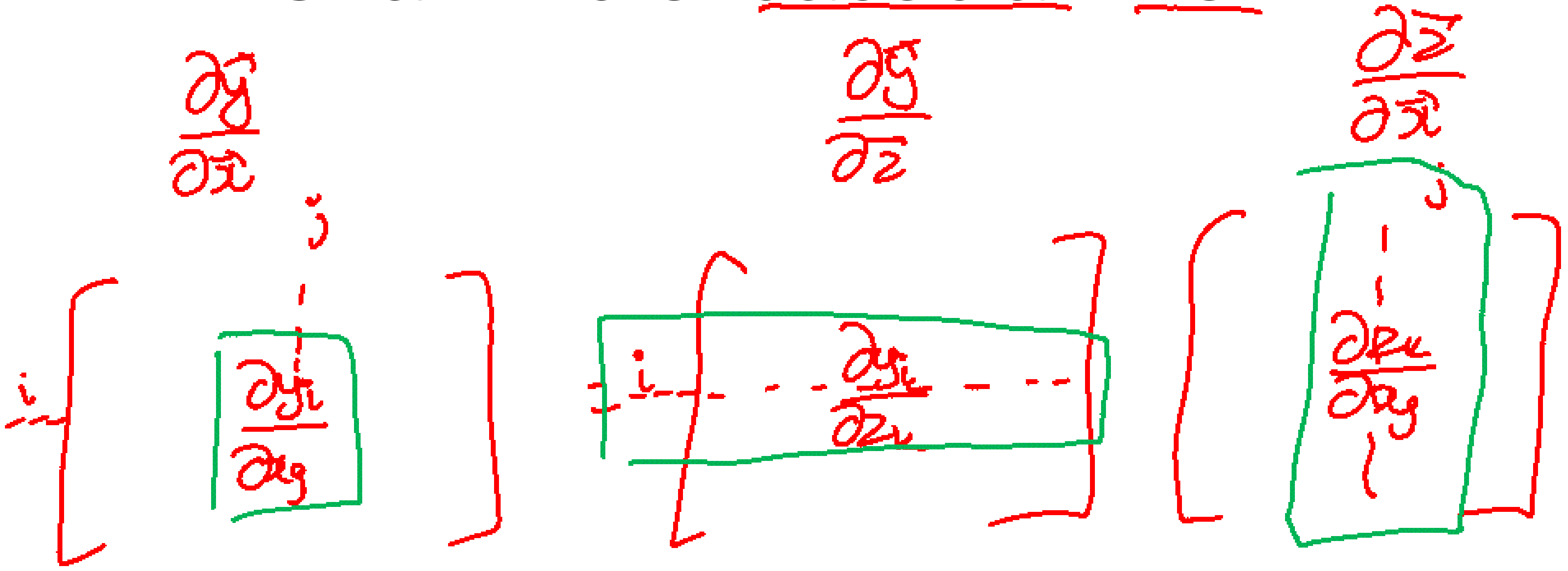
Chain Rule: Vector Case

$$\vec{x} \in \mathbb{R}^d \xrightarrow[:\mathbb{R}^d \to \mathbb{R}^m]{g_1(\,)} \vec{z} \in \mathbb{R}^m \xrightarrow[:\mathbb{R}^m \to \mathbb{R}^c]{g_2(\,)} \vec{y} \in \mathbb{R}^c$$

$$\underbrace{\left[\frac{\partial \vec{y}}{\partial \vec{x}}\right]}_{\boxed{J_{g_2 \circ g_1}}} = \underbrace{\left[\frac{\partial \vec{y}}{\partial \vec{z}}\right]}_{J_{g_2}} \odot \underbrace{\left[\frac{\partial \vec{z}}{\partial \vec{x}}\right]}_{J_{g_1}}$$

Matrix Mult

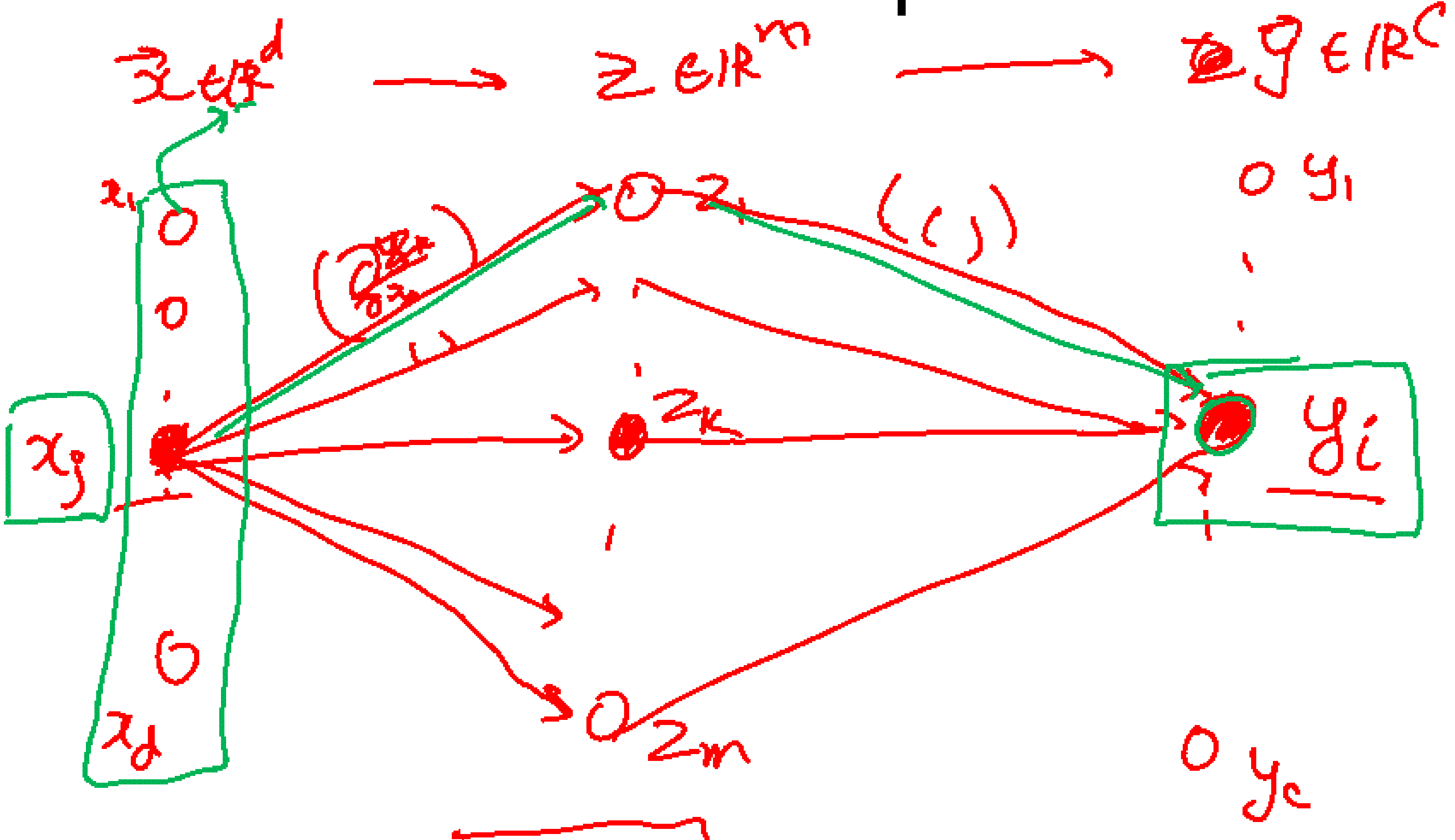
Chain Rule: Jacobian view



$\frac{\partial y}{\partial x}$
j
i
$\frac{\partial y_i}{\partial x_j}$
$\frac{\partial y}{\partial z}$
i
$\frac{\partial y_i}{\partial z_k}$
$\frac{\partial z}{\partial x}$
j
$\frac{\partial z_k}{\partial x_j}$

$$\frac{\partial y_i}{\partial x_j} = \sum_{\boxed{k}} \frac{\partial y_i}{\partial z_k} \frac{\partial z_k}{\partial x_j}$$

Chain Rule: Graphical view



$$\frac{\partial y_i}{\partial x_j} = \boxed{\sum_{k \text{ on path}}} \frac{\partial y_i}{\partial z_k} \frac{\partial z_k}{\partial x_j}$$

Chain Rule: Cascaded

$$x = \vec{h}^0 \xrightarrow{g_1} \vec{h}^1 \in \mathbb{R}^d \xrightarrow{g_2(\cdot)} \vec{h}^{(2)} \in \mathbb{R}^d \cdots \xrightarrow{g_\ell(\cdot)} \vec{h}^{(\ell)} \to L$$

$$\frac{\partial L}{\partial \vec{h}^{(1)}} = \boxed{\frac{\partial L}{\partial \vec{h}^{\ell}}}\ \boxed{\frac{\partial \vec{h}^{(\ell)}}{\partial \vec{h}^{(1)}}} = \boxed{\frac{\partial L}{\partial \vec{h}^{(\ell)}}}\ \boxed{\frac{\partial \vec{h}^{(\ell)}}{\partial \vec{h}^{\ell-1}}}\ \boxed{\frac{\partial \vec{h}^{\ell-1}}{\partial \vec{h}^{(\ell-2)}}} \cdots \boxed{\frac{\partial \vec{h}^{(2)}}{\partial \vec{h}^{(1)}}}$$

$$[\ ] = [\ \ ]\,[\ \ ]_{d \times d} = \underbrace{[\ \ ]_{1 \times d}\,[\ \ ]_{d\times d}}_{O(\ell d^2)} \longleftrightarrow \underbrace{[\ \ ]_{d\times d}\,[\ \ ]_{d\times d}}_{O(\ell d^3)}$$

Deep Learning = Differentiable Programming

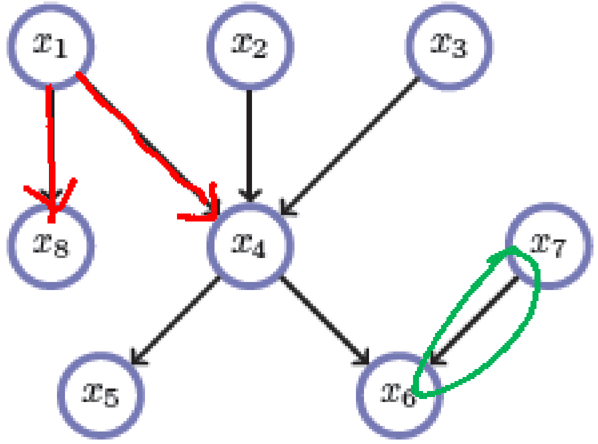
- Computation = $\boxed{\text{Graph}}$
  - Input = $\overset{x,y}{\text{Data}}$ + Parameters w
  - Output = $\boxed{\text{Loss}}$
  - Scheduling = Topological ordering

- Auto-Diff
  - A family of algorithms for implementing chain-rule on computation graphs

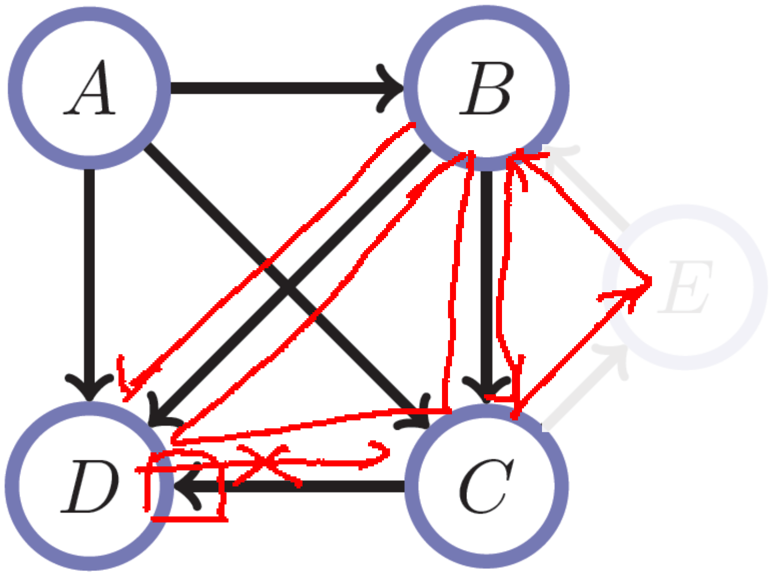
Directed Acyclic Graphs (DAGs)

- Exactly what the name suggests
  - Directed edges
  - No (directed) cycles
  - Underlying undirected cycles okay

DAG ✓



✗



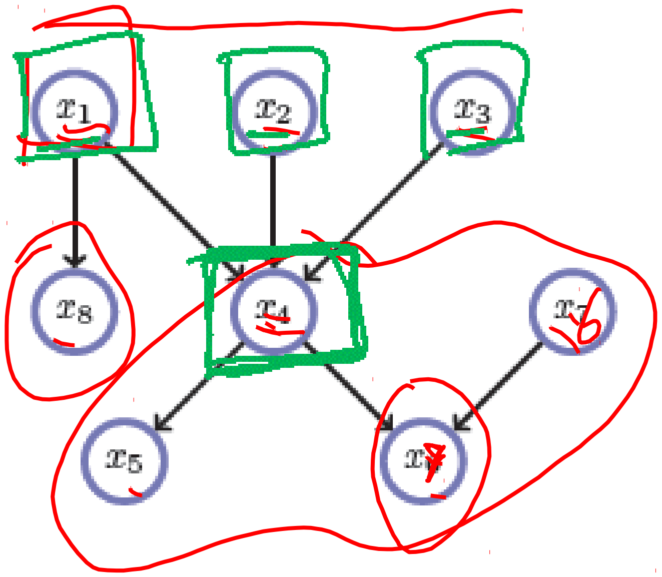
Directed Acyclic Graphs (DAGs)

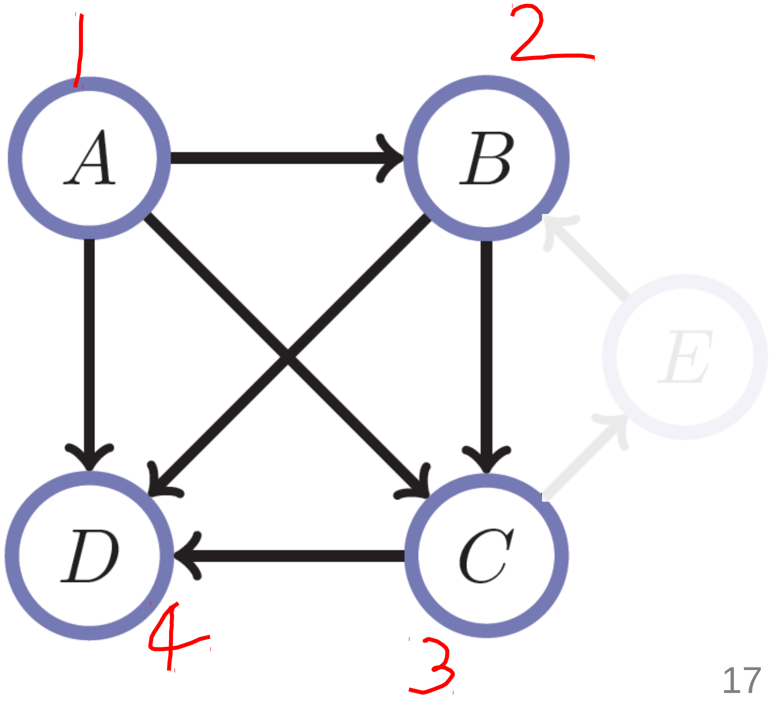
- Concept
  - Topological Ordering

$\exists$ bijection $\sigma : V \rightarrow \{1, \ldots, n\}$

s.t. $\forall (v_i, v_j) \in E$

$\sigma(v_i) < \sigma(v_j)$



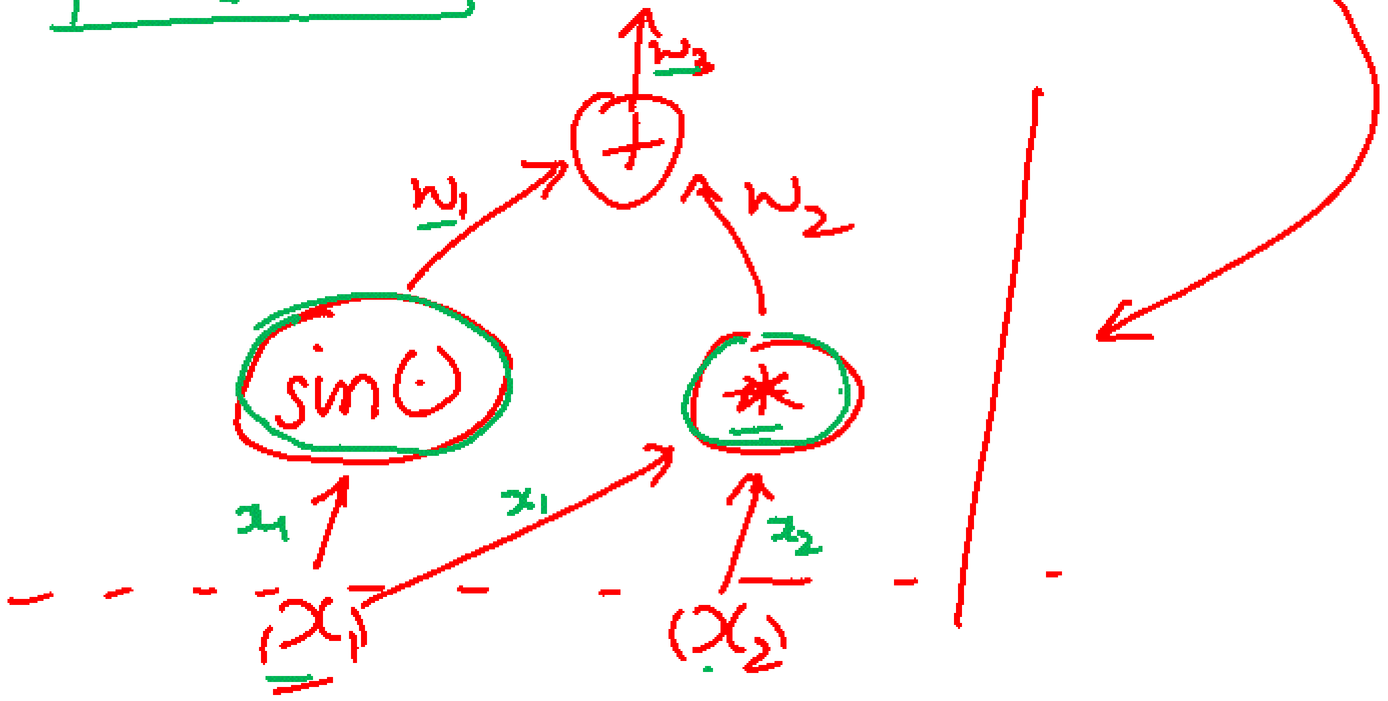


Computational Graphs ← DAG

- Notation

$$\boxed{f(\underline{x_1}, \underline{x_2})} = \underbrace{x_1 x_2}_{} + \underbrace{\sin(x_1)}_{}$$

w_3 w_1



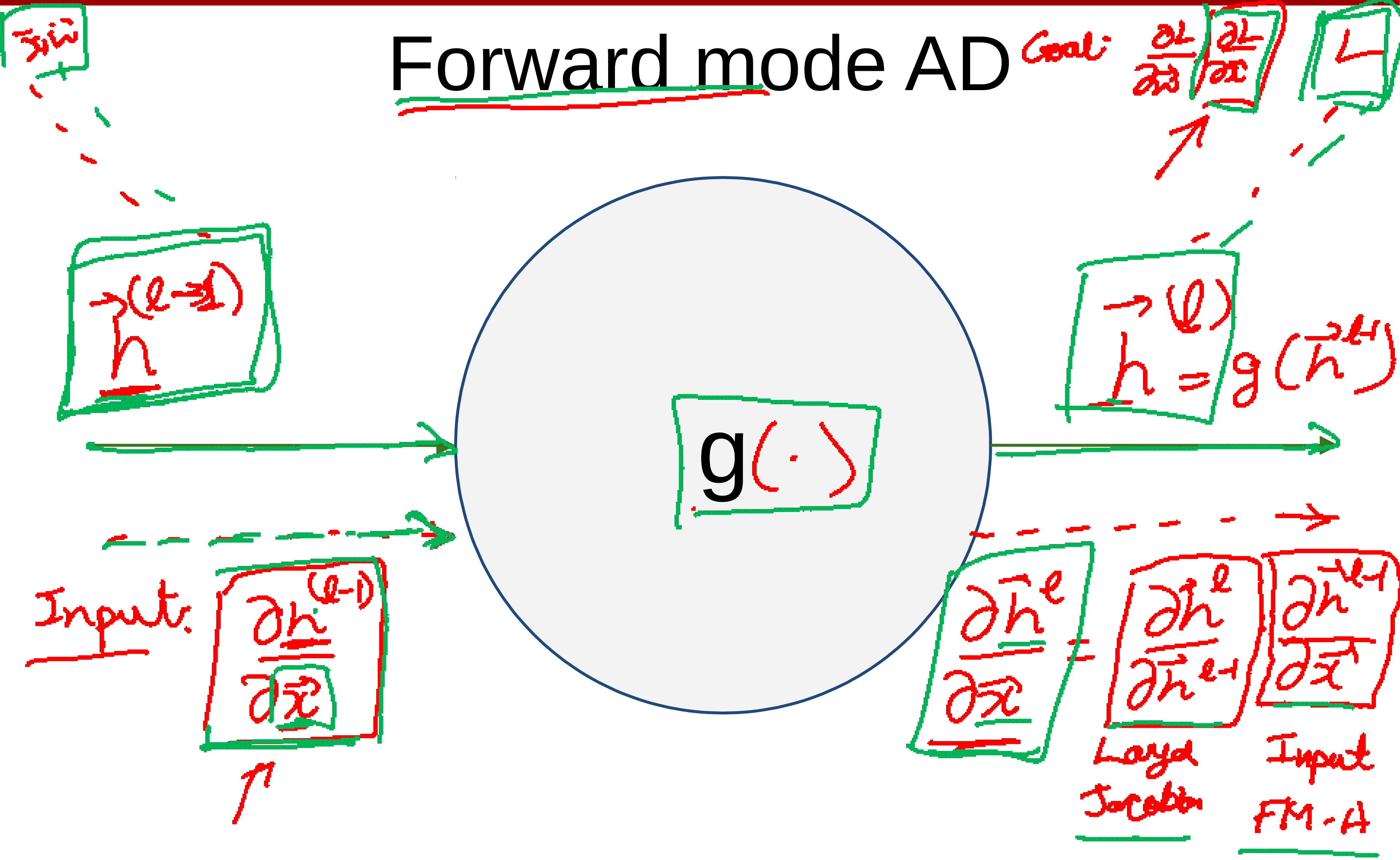
Deep Learning = Differentiable Programming

- Computation = Graph
  - Input = Data + Parameters
  - Output = Loss
  - Scheduling = Topological ordering

- Auto-Diff
  - A family of algorithms for implementing chain-rule on computation graphs

Forward mode AD

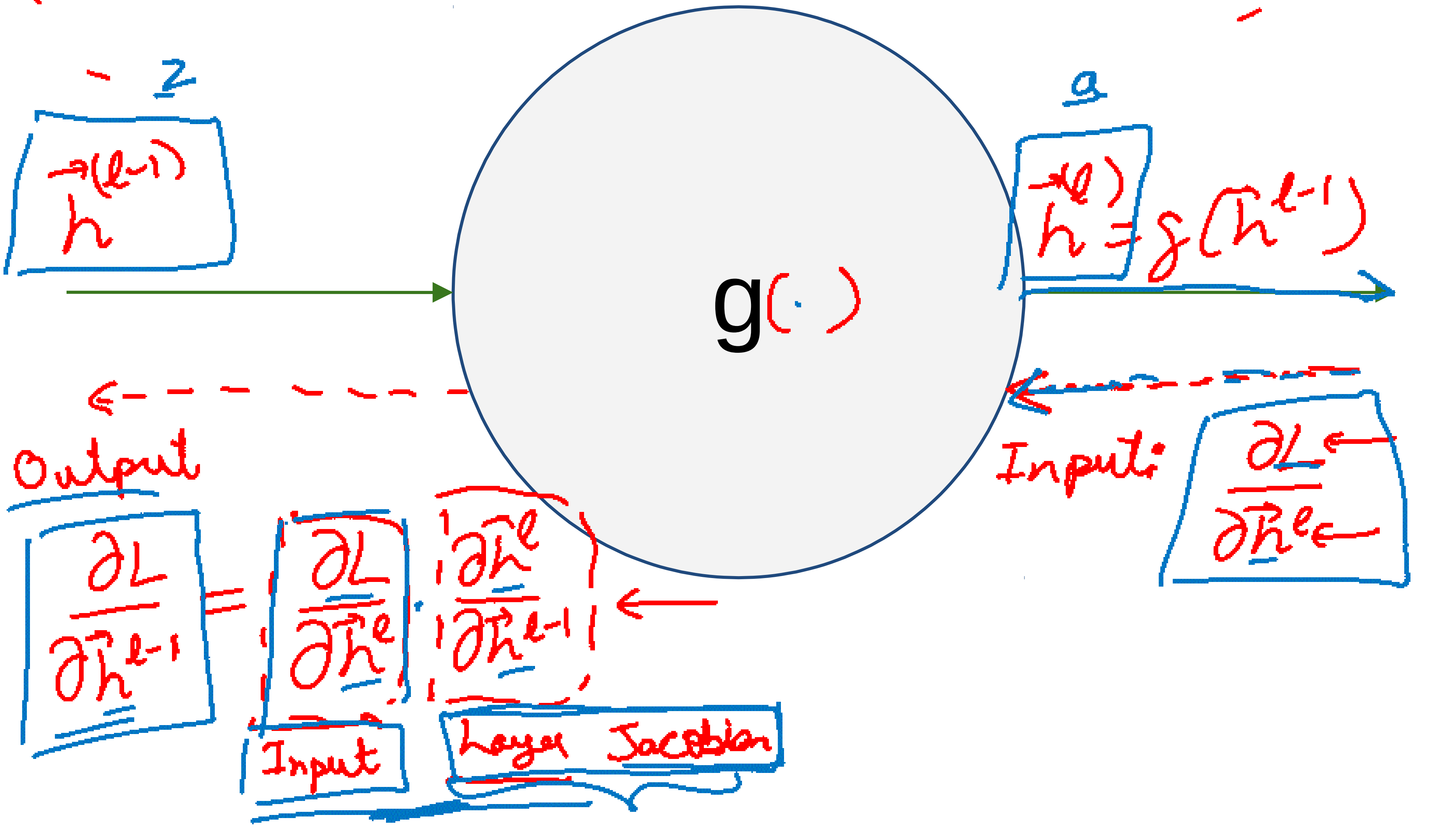
Goal: $\frac{\partial L}{\partial \vec{x}}$



$\vec{x}_i$
$\vec{h}^{(\ell-1)}$
$g(\cdot)$
$\vec{h}^{(\ell)} = g(\vec{h}^{\ell-1})$
L
Input:
$\frac{\partial \vec{h}^{(\ell-1)}}{\partial \vec{x}}$
$\frac{\partial \vec{h}^{\ell}}{\partial \vec{x}} = \frac{\partial \vec{h}^{\ell}}{\partial \vec{h}^{\ell-1}} \frac{\partial \vec{h}^{\ell-1}}{\partial \vec{x}}$
Layer Jacobian
Input FM-A

Reverse mode AD

Goal: $\frac{\partial L}{\partial x}$



$g(\cdot)$
$\vec{h}^{(l-1)}$
$\vec{h}^{(l)} = g(\vec{h}^{l-1})$
Input: $\frac{\partial L}{\partial \vec{h}^{l}}$
Output
$\frac{\partial L}{\partial \vec{h}^{l-1}} = \frac{\partial L}{\partial \vec{h}^{l}} \cdot \frac{\partial \vec{h}^{l}}{\partial \vec{h}^{l-1}}$
Input
Layer Jacobian

Plan for Today

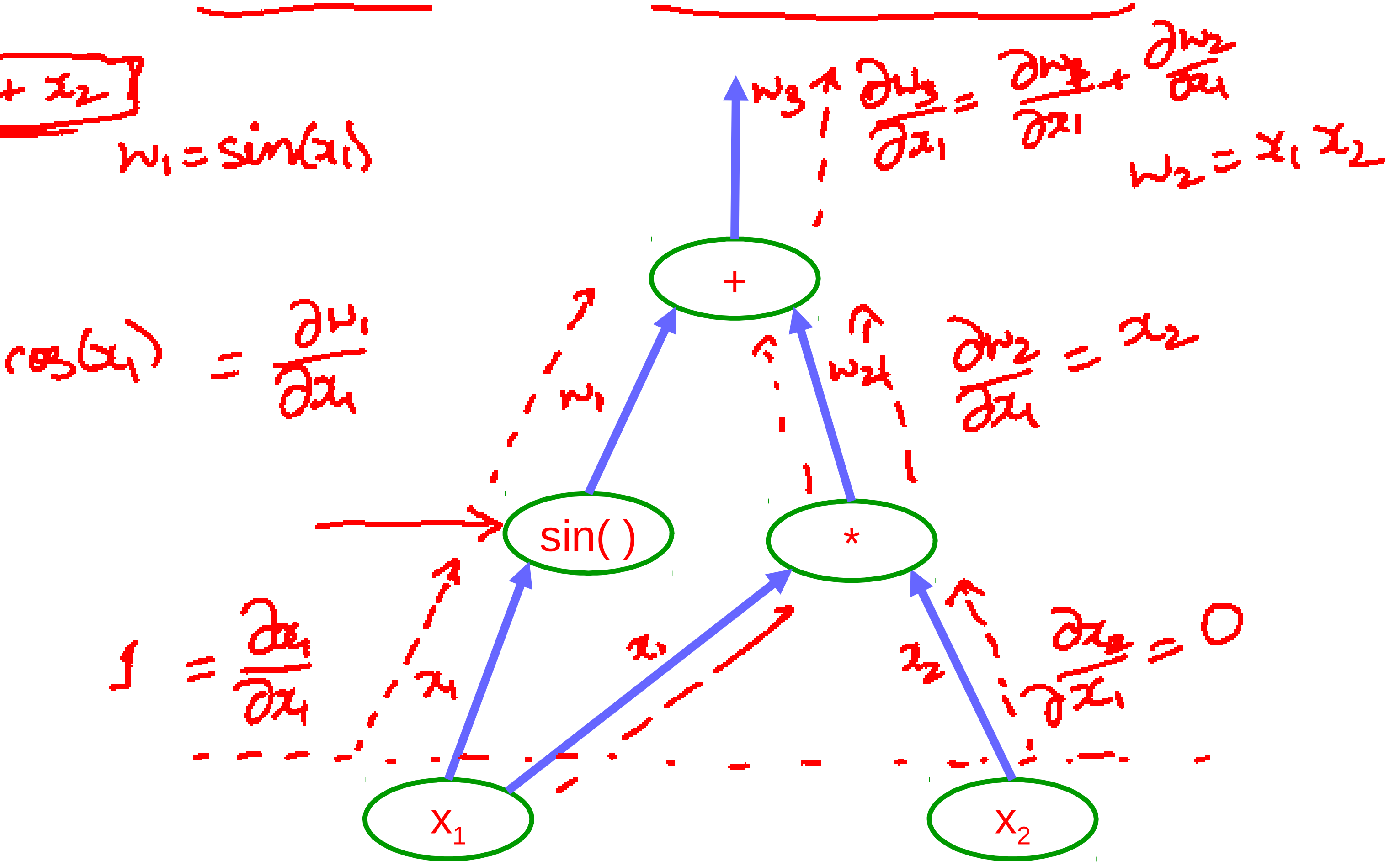
- Automatic Differentiation
  - (Finish) Forward mode vs Reverse mode AD
  - Backprop
  - Patterns in backprop

Example: Forward mode AD

$$\underline{f(x_1, x_2)} = \underline{\overbrace{\sin(x_1)}^{w_1} + \overbrace{x_1 x_2}^{w_2}}$$

$$\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1} \; \frac{\partial f}{\partial x_2}\right]$$

$$\boxed{\frac{\partial f}{\partial x_1}} = \cos x_1 + x_2$$



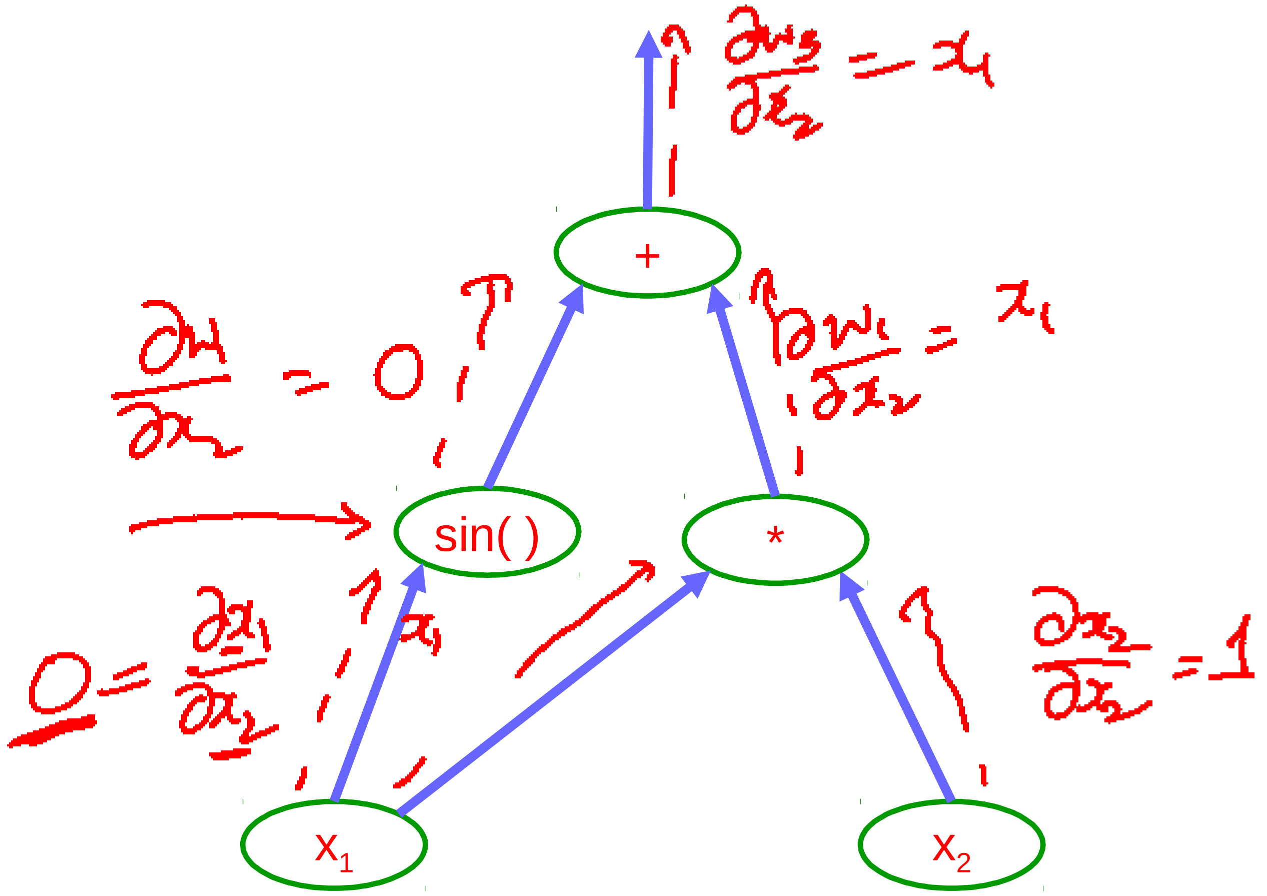
$w_1 = \sin(x_1)$
$w_2 = x_1 x_2$
w_3
$\frac{\partial w_3}{\partial x_1} = \frac{\partial w_1}{\partial x_1} + \frac{\partial w_2}{\partial x_1}$
+
w_1
w_2
$\cos(x_1) = \frac{\partial w_1}{\partial x_1}$
$\frac{\partial w_2}{\partial x_1} = x_2$
sin( )
*
x_1
x_1
x_2
$1 = \frac{\partial x_1}{\partial x_1}$
$\frac{\partial x_2}{\partial x_1} = 0$
x_1
x_2

Example: Forward mode AD

$$f(x_1, x_2) = \cancel{\sin(x_1)} + \cancel{x_1 x_2}$$

$\boxed{\frac{\partial f}{\partial x_2}}$

$= x_1$



$\frac{\partial w_3}{\partial x_2} = x_1$
+
$\frac{\partial w_1}{\partial x_2} = 0$
$\frac{\partial w_1}{\partial x_2} = x_1$
sin( )
*
x_1
$0 = \frac{\partial x_1}{\partial x_2}$
$\frac{\partial x_2}{\partial x_2} = 1$
X$_1$
X$_2$

Example: Forward mode AD

$$\frac{\partial f}{\partial x} = \left[\boxed{\frac{\partial f}{\partial x_1}}\ \boxed{\frac{\partial f}{\partial x_2}}\right]$$

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

for a in $\{x_1, x_2\}$

compute/evaluate

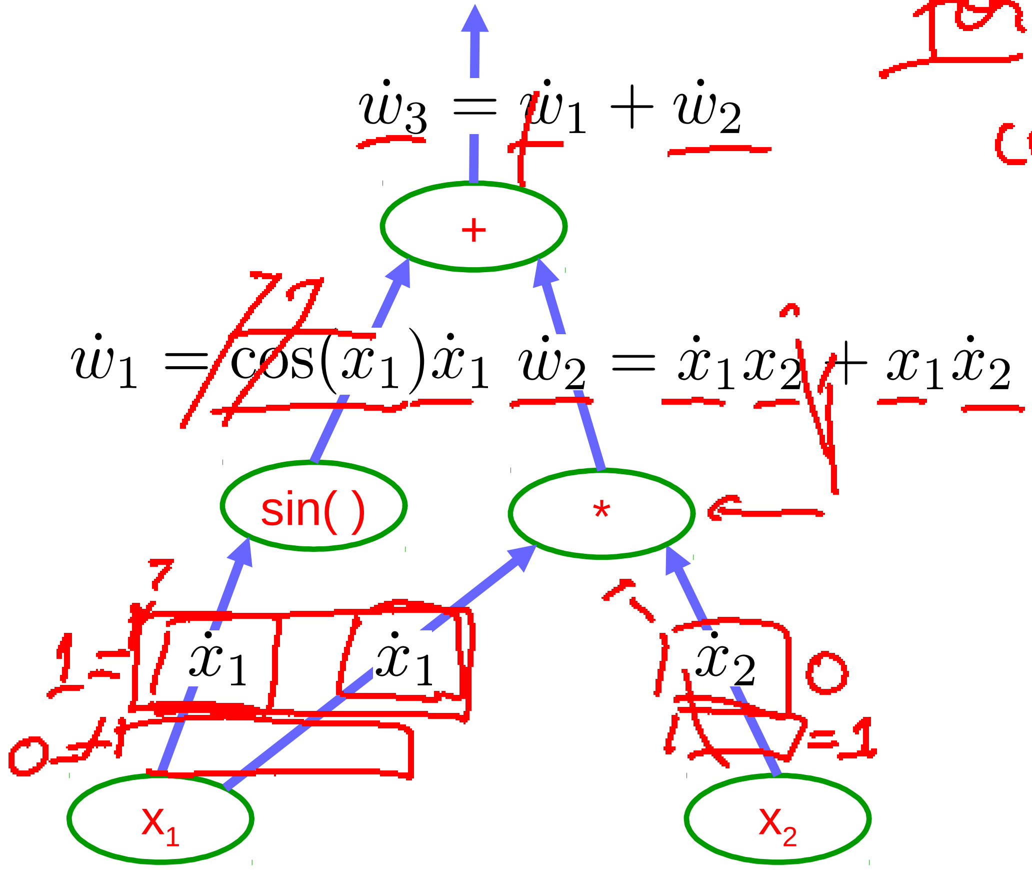
$\dot{w}_3 = \frac{\partial w_3}{\partial a}$

$\dot{w}_2 = \frac{\partial w_2}{\partial a}$

$\dot{w}_1 = \frac{\partial w_1}{\partial a}$

$\dot{x}_1 = \frac{\partial x_1}{\partial a}$

$\dot{x}_2 = \frac{\partial x_2}{\partial a}$

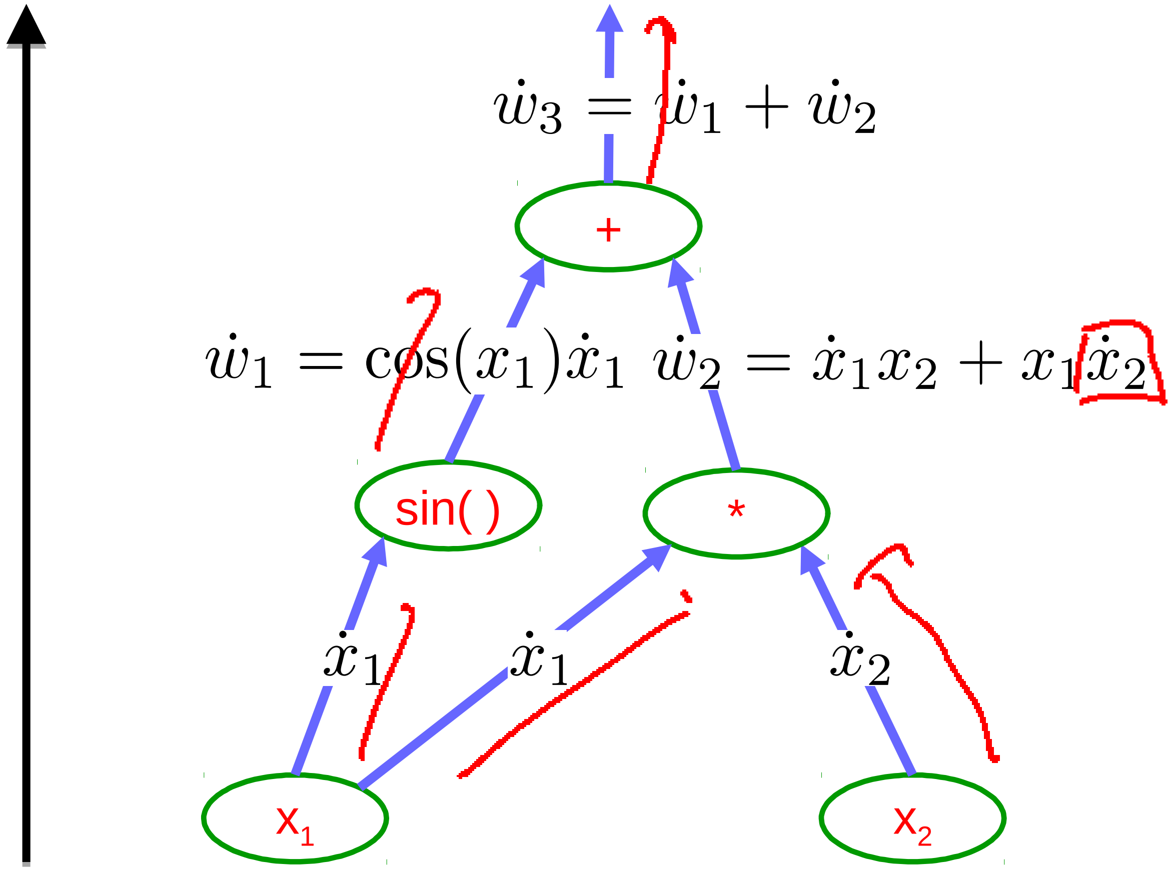


$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
+
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
sin( )
*
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
1 =
0 =
0
=1
X1
X2

Example: Forward mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

$$\frac{\partial f}{\partial x}$$

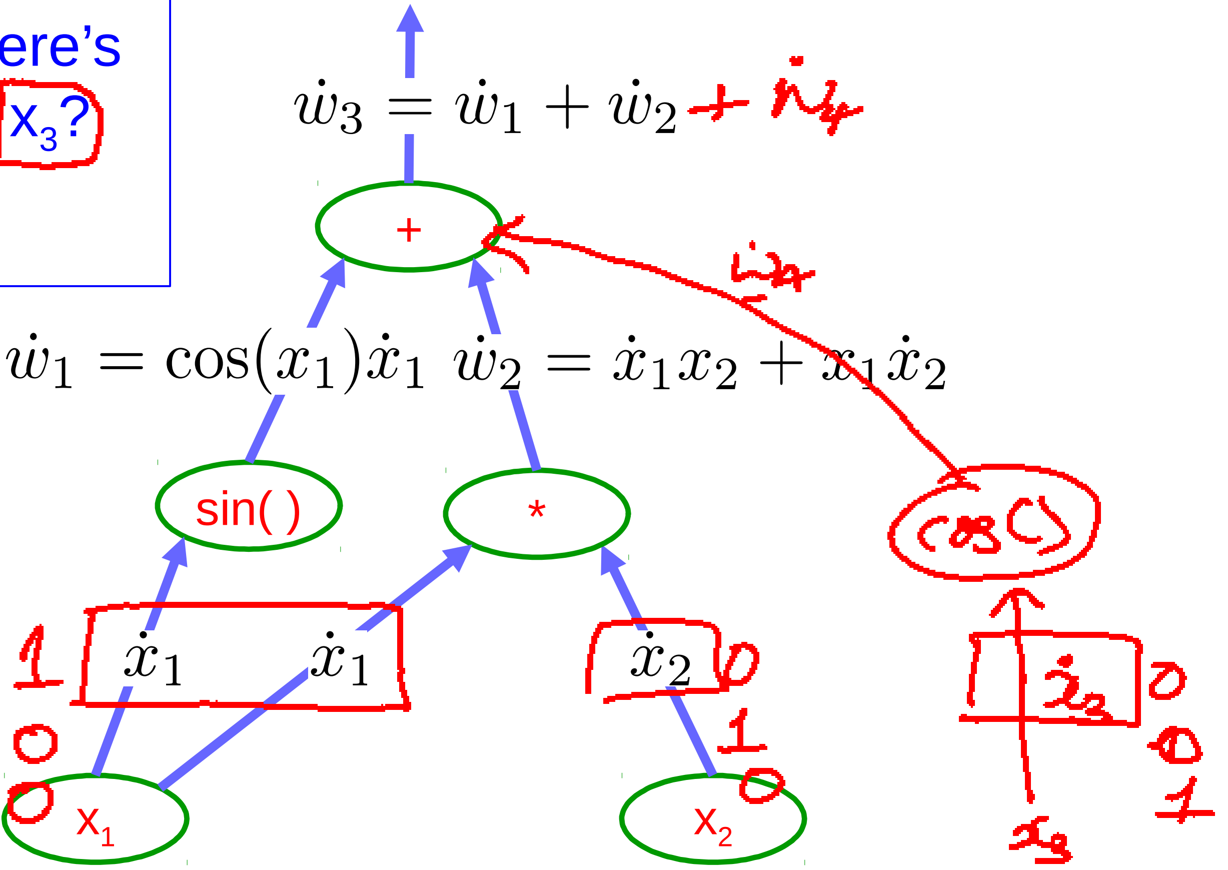

$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
+
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \boxed{\dot{x}_2}$
sin( )
*
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
x_1
x_2

Example: Forward mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2 + \cos(x_3)$$

Q: What happens if there's another input variable $\boxed{x_3}$?

for a in $\{x_1, x_2, x_3\}$



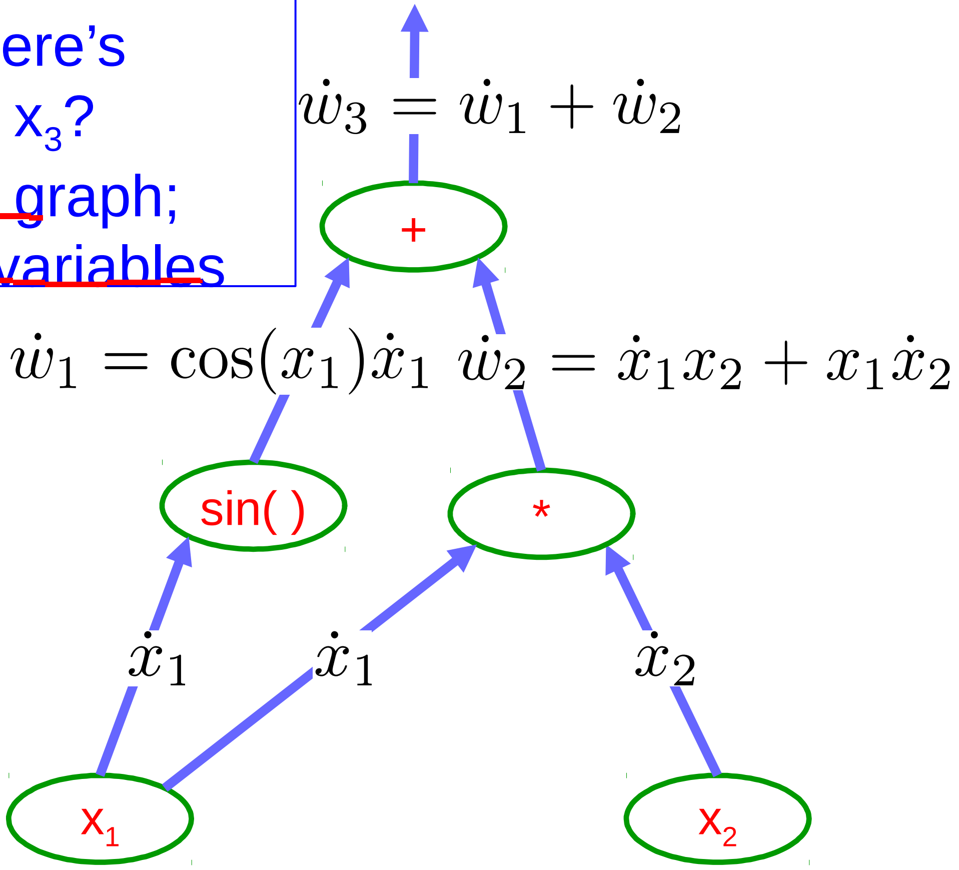
$\dot{w}_3 = \dot{w}_1 + \dot{w}_2 + \dot{w}_4$
+
$\dot{w}_4$
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
sin( )
*
cos( )
1
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
0
$\dot{x}_3$
0
0
1
0
0
1
x_1
x_2
x_3

Example: Forward mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

Q: What happens if there's another input variable x_3?

A: more sophisticated graph; d+1 "passes" for d+1 variables



$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
+
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
sin( )
*
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
x_1
x_2

Example: Forward mode AD

$\frac{\partial \vec{f}}{\partial \vec{x}}$

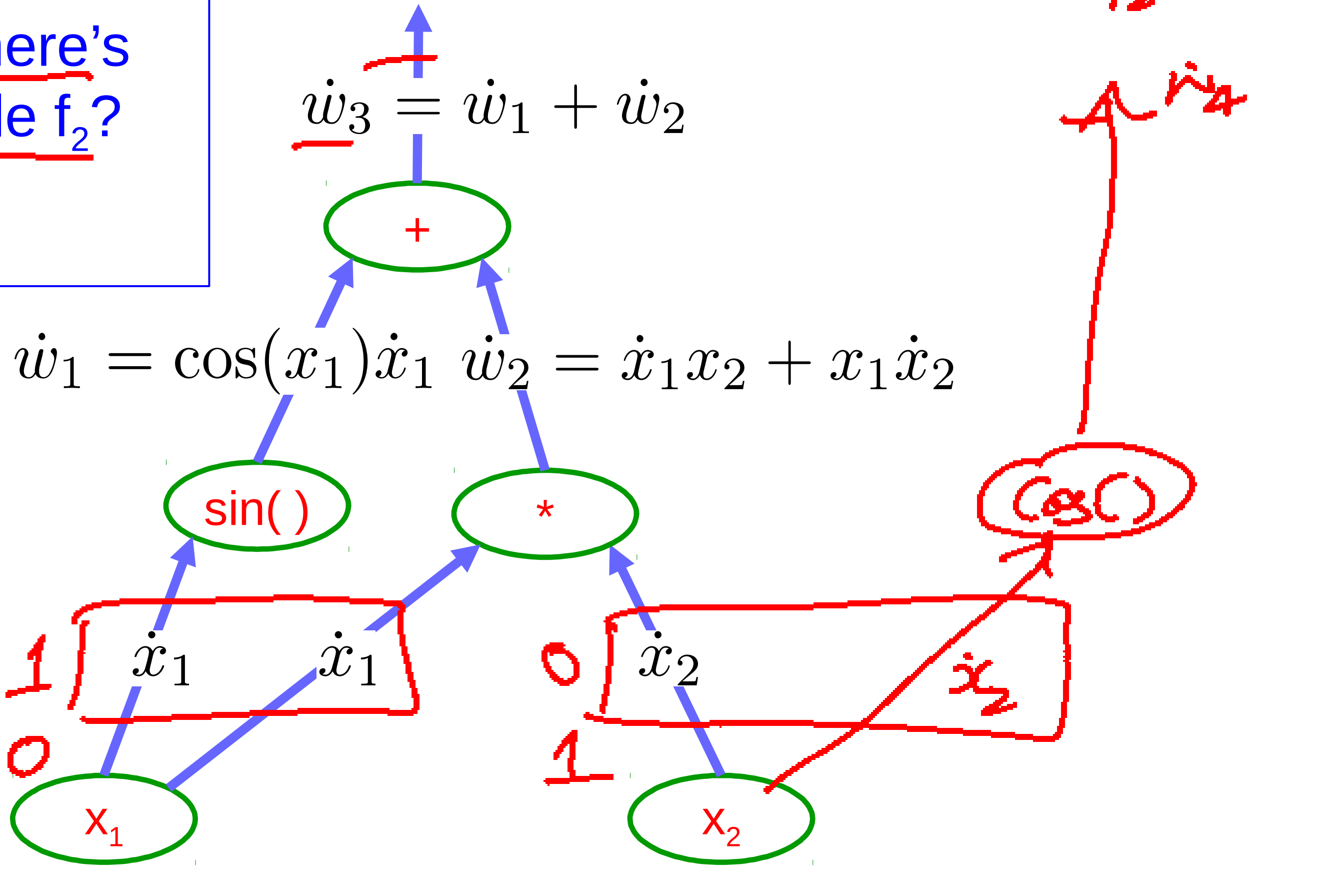
$$f_1(x_1, x_2) = \sin(x_1) + x_1 x_2$$

$f_2 = \cos(x_2)$

$f_2 = w_4$

Q: What happens if there's another output variable f_2?

for a in $\{x_1, x_2\}$



$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
+
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
sin( )
*
cos()
$\dot{w}_4$
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
$\dot{x}_2$
1
0
0
1
x_1
x_2

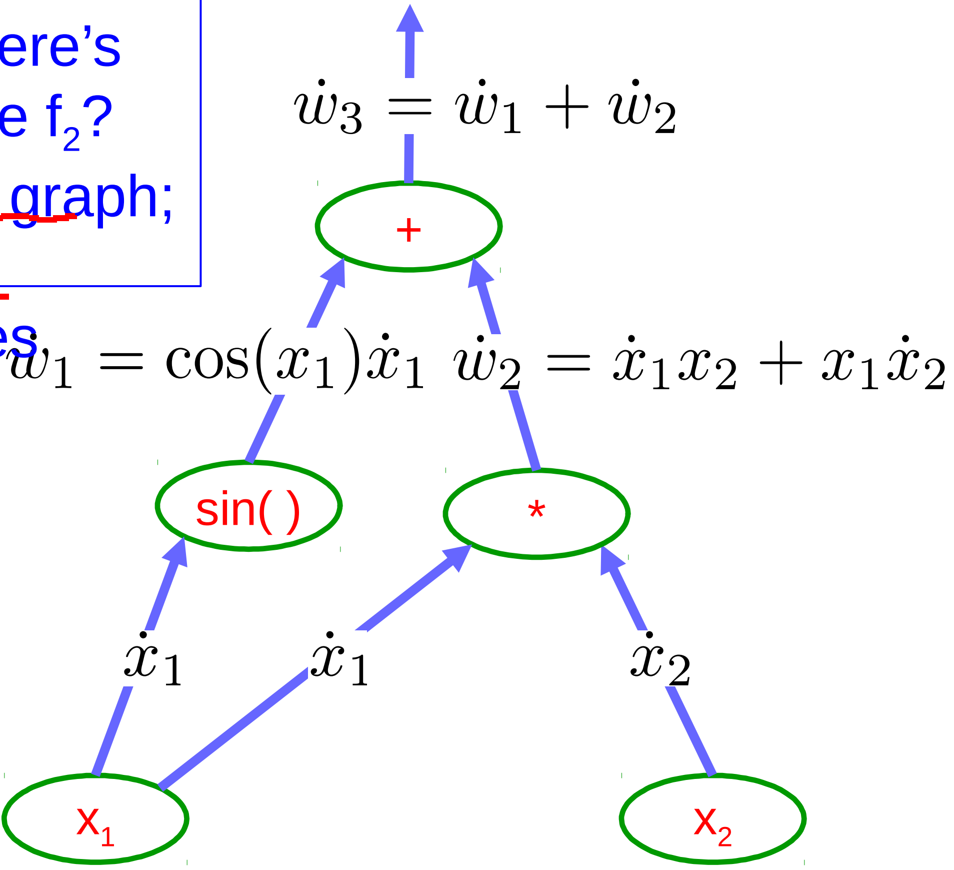
Example: Forward mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

Q: What happens if there's another output variable f_2?

A: ~~more sophisticated graph;~~ [illegible]

d "passes" for variables



$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
+
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
sin( )
*
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
x_1
x_2

Example: Reverse mode AD

Goal: $\frac{\partial f}{\partial x} = \left[\frac{\partial f}{\partial x_1} \; \frac{\partial f}{\partial x_2}\right]$

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

$w_3 = w_1 + w_2$

$\frac{\partial w_3}{\partial w_1} = 1$

$\bar{w}_1 = \frac{\partial f}{\partial w_1} = \boxed{\frac{\partial f}{\partial w_3}} \boxed{\frac{\partial w_3}{\partial w_1}}$

big def.

$\bar{w}_3 = \boxed{\frac{\partial f}{\partial w_3}}$

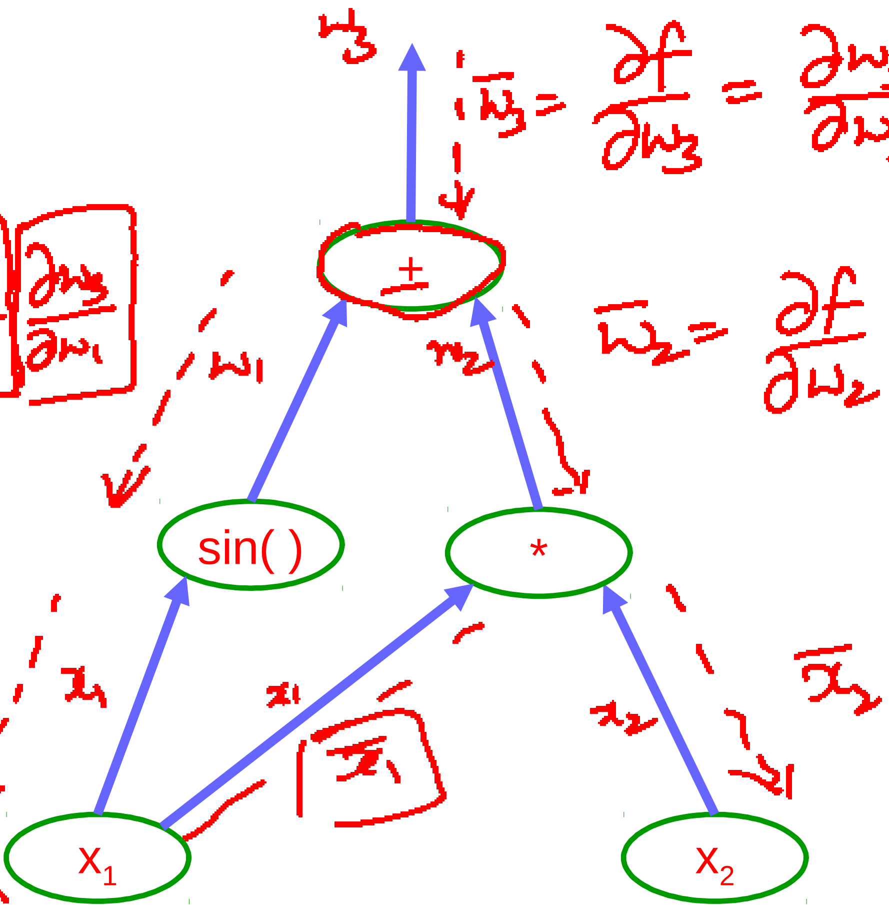
$\boxed{\bar{x}_1} = \frac{\partial f}{\partial x_1} = \boxed{\frac{\partial f}{\partial w_1}} \boxed{\frac{\partial w_1}{\partial x_1}}$

input $\bar{w}_1$

Jacobian $(\cos(x_1))$

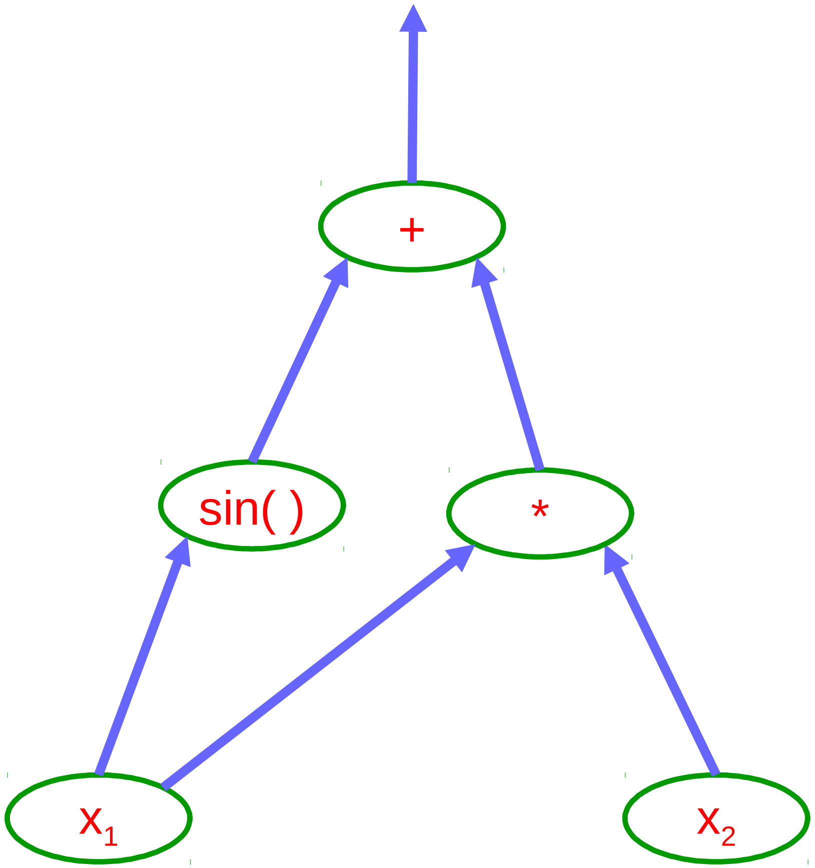
$\bar{w}_3 = \frac{\partial f}{\partial w_3} = \frac{\partial w_3}{\partial w_3} = 1$

$\bar{w}_2 = \frac{\partial f}{\partial w_2} = \boxed{\frac{\partial f}{\partial w_3}}$

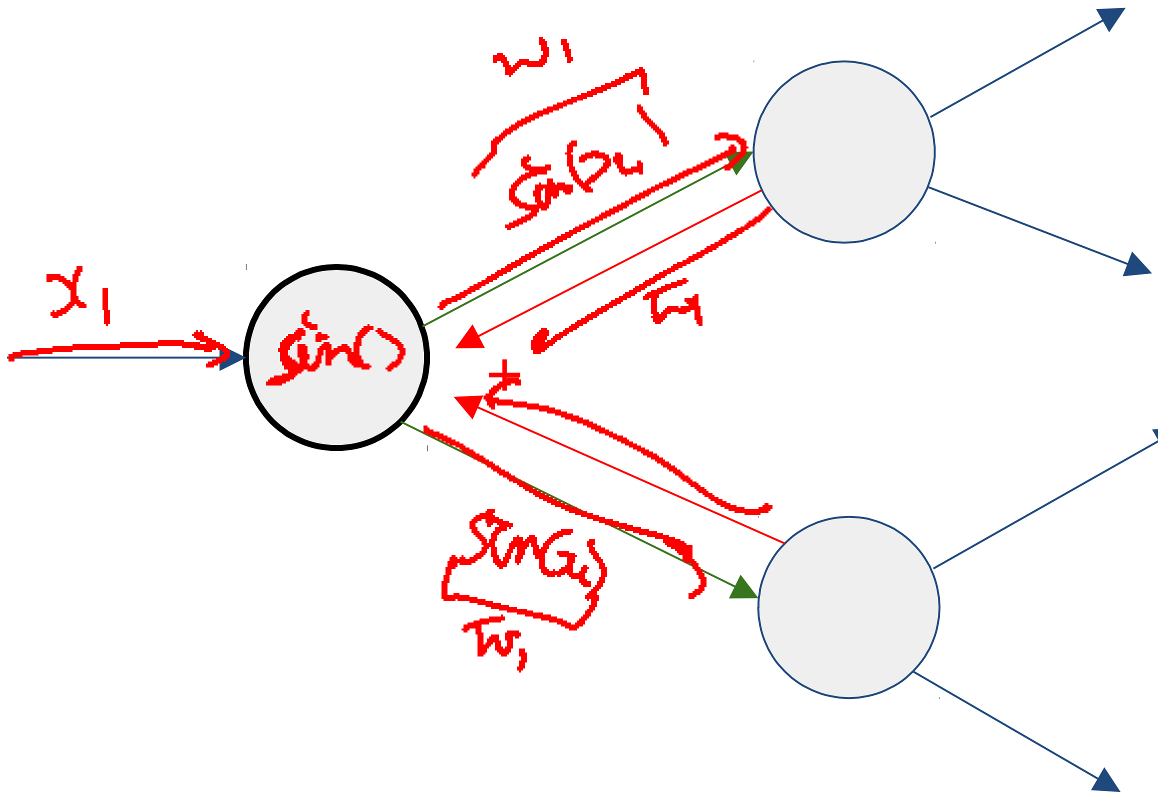


Example: Reverse mode AD

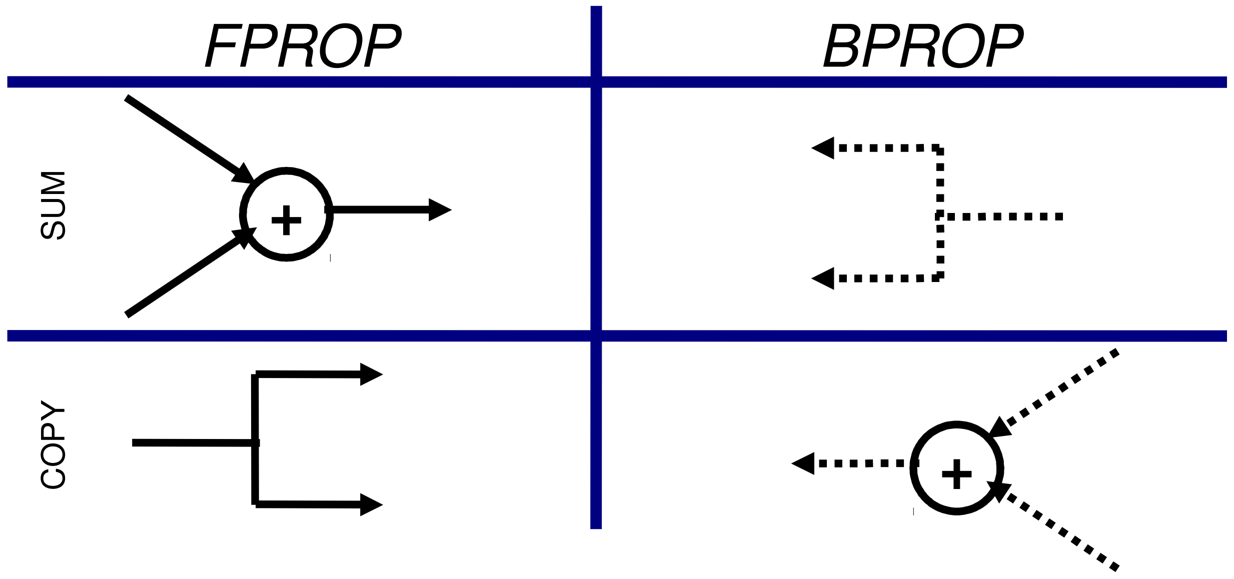
$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$



Gradients add at branches

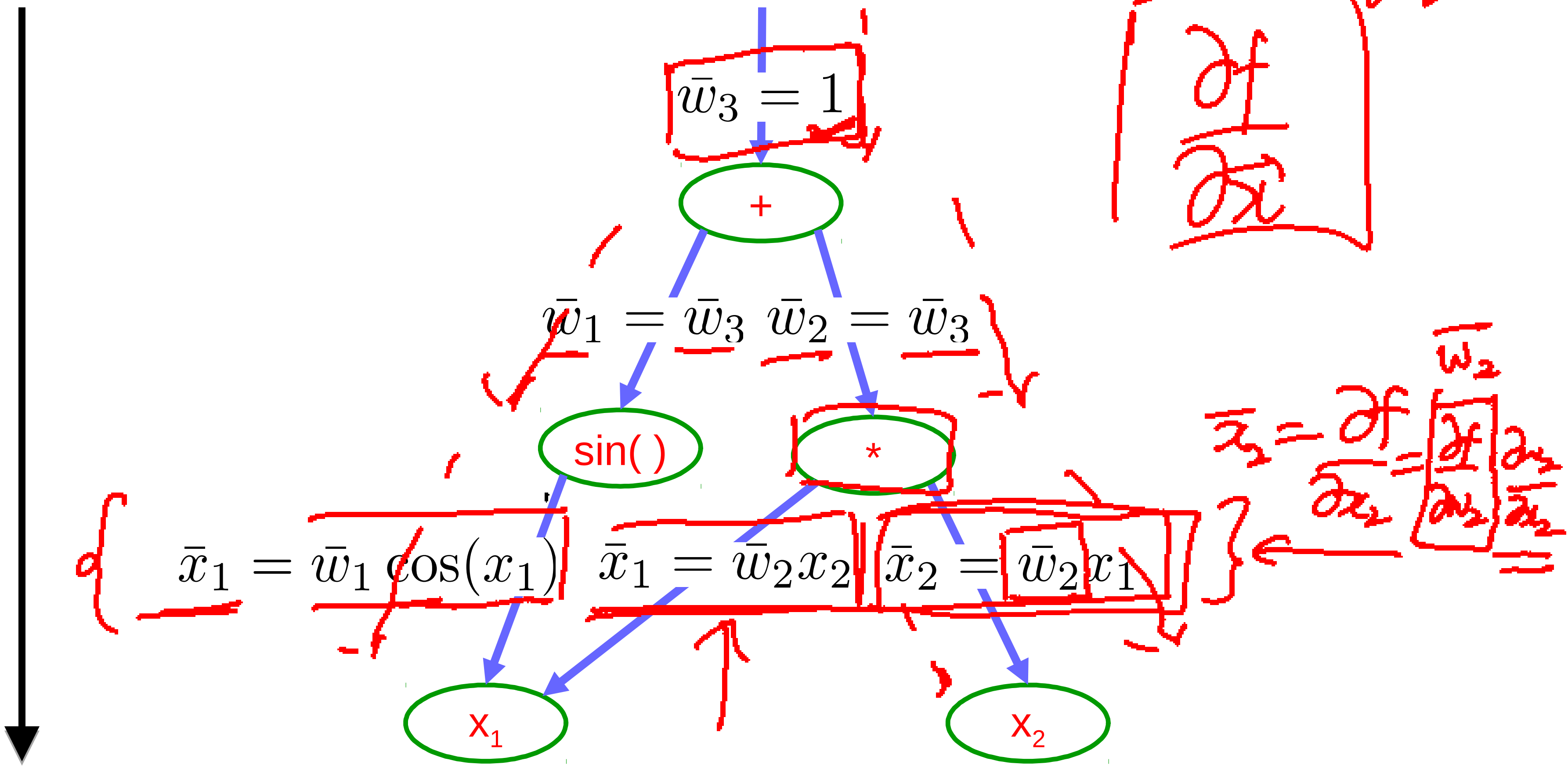


Duality in Fprop and Bprop



Example: Reverse mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$



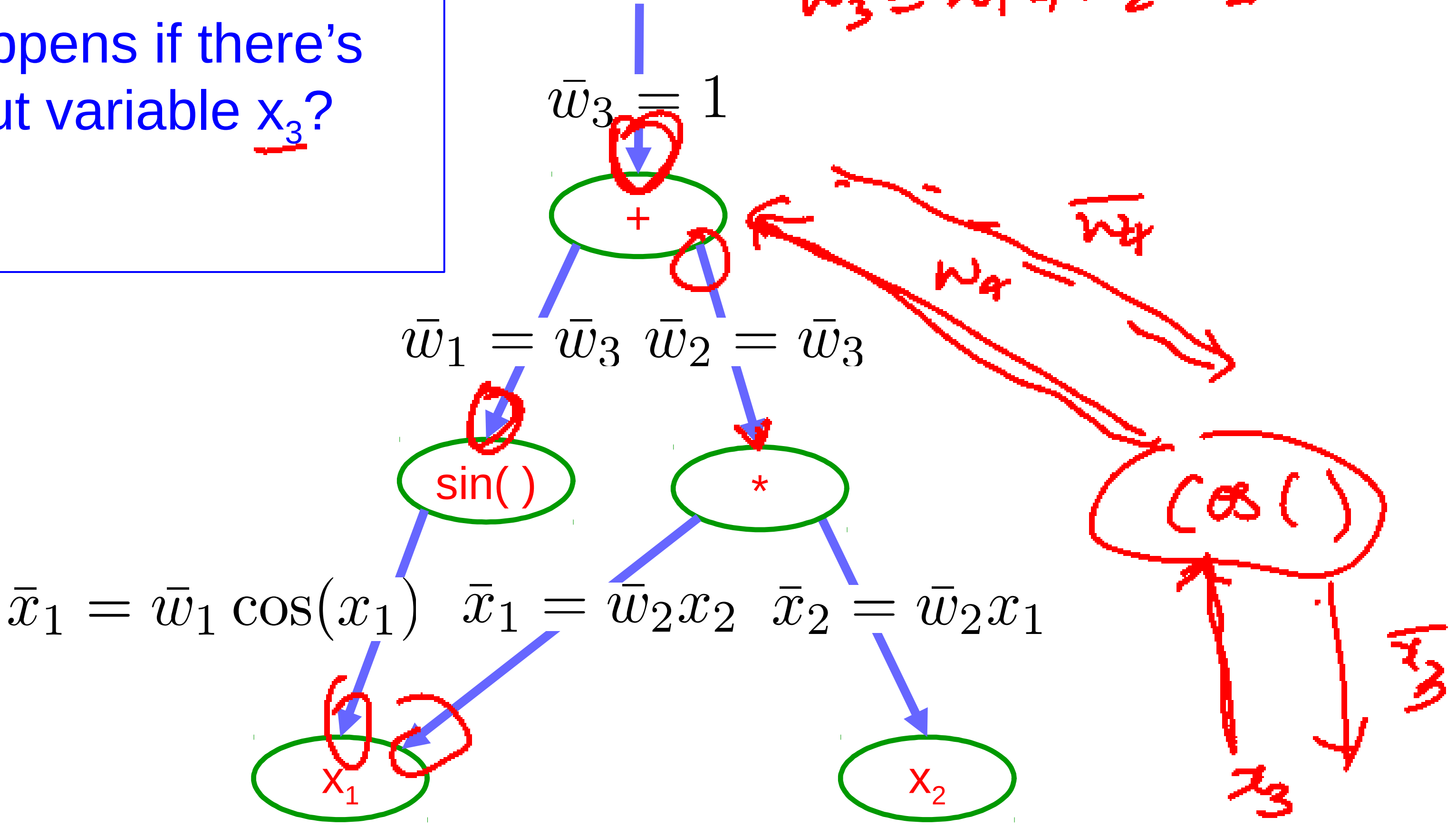
$\bar{w}_3 = 1$
+
$\bar{w}_1 = \bar{w}_3$
$\bar{w}_2 = \bar{w}_3$
sin( )
*
$\bar{x}_1 = \bar{w}_1 \cos(x_1)$
$\bar{x}_1 = \bar{w}_2 x_2$
$\bar{x}_2 = \bar{w}_2 x_1$
x_1
x_2
$w_2 = x_1 x_2$
$\frac{\partial w_2}{\partial x_2} = x_1$
$\frac{\partial f}{\partial x}$
$\bar{w}_2$
$\bar{x}_2 = \frac{\partial f}{\partial x_2} = \frac{\partial f}{\partial w_2} \frac{\partial w_2}{\partial x_2}$

Example: Reverse mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2 + \cos(x_3)$$

$w_3 = w_1 + w_2 + w_4$

Q: What happens if there's another input variable x_3?



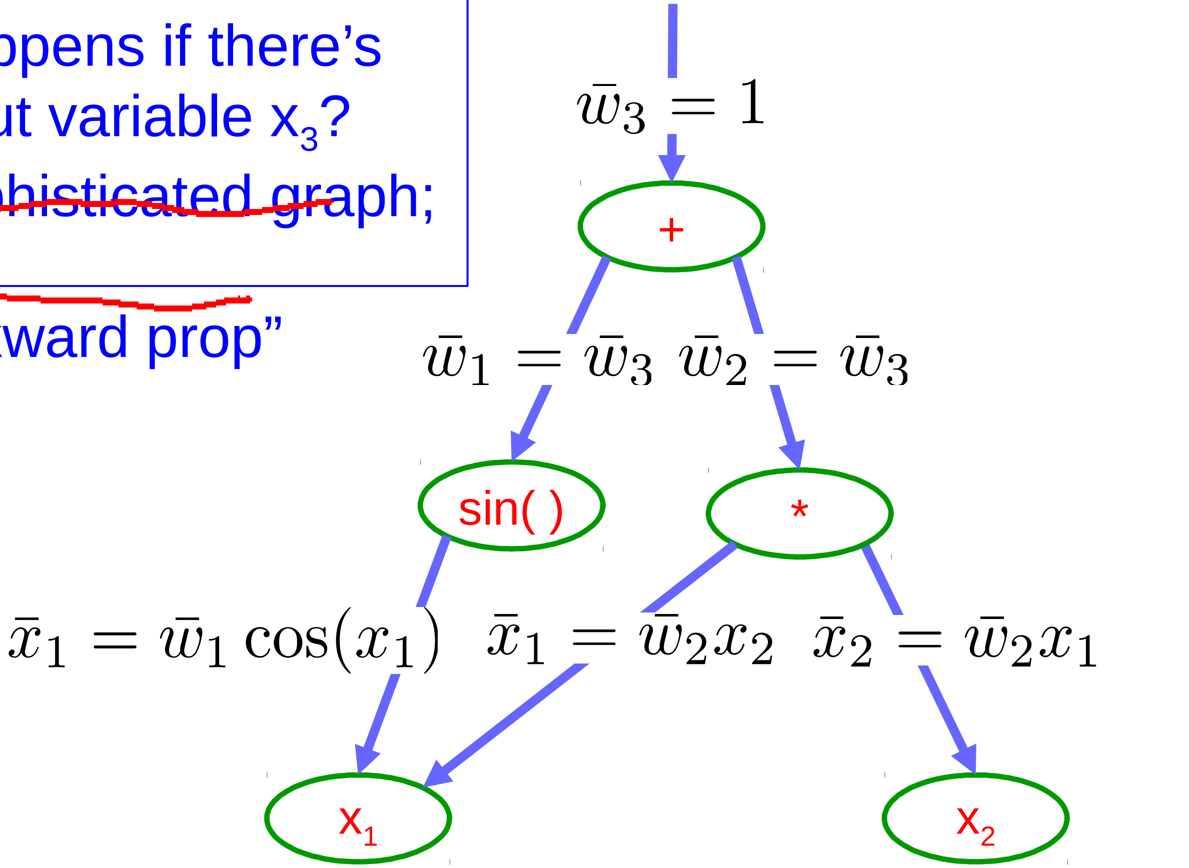
Example: Reverse mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

Q: What happens if there's another input variable x_3?

A: ~~more sophisticated graph~~;

single "backward prop"

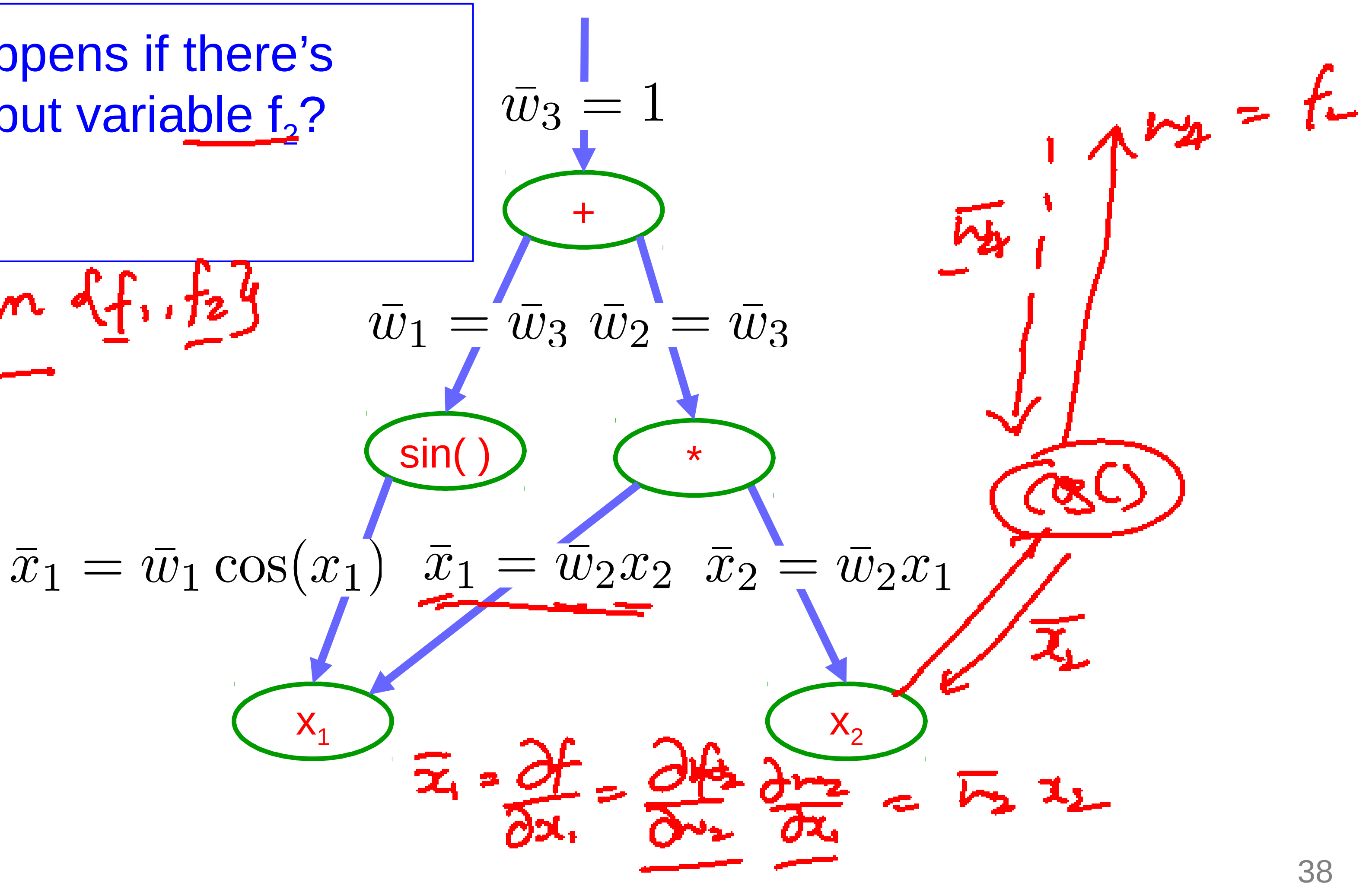


$\bar{w}_3 = 1$
+
$\bar{w}_1 = \bar{w}_3$
$\bar{w}_2 = \bar{w}_3$
sin( )
*
$\bar{x}_1 = \bar{w}_1 \cos(x_1)$
$\bar{x}_1 = \bar{w}_2 x_2$
$\bar{x}_2 = \bar{w}_2 x_1$
x_1
x_2

Example: Reverse mode AD

$$f_1(x_1, x_2) = \sin(x_1) + x_1 x_2 \qquad f_2 = \cos(x_2)$$

Q: What happens if there's another output variable f_2?



for a in $\{f_1, f_2\}$

$\bar{w}_4 = \frac{\partial a}{\partial w_4}$

$\bar{w}_3 = \frac{\partial a}{\partial w_3}$

$$\bar{x}_1 = \frac{\partial f}{\partial x_1} = \frac{\partial f_2}{\partial w_2}\frac{\partial w_2}{\partial x_1} = \bar{w}_2 x_2$$

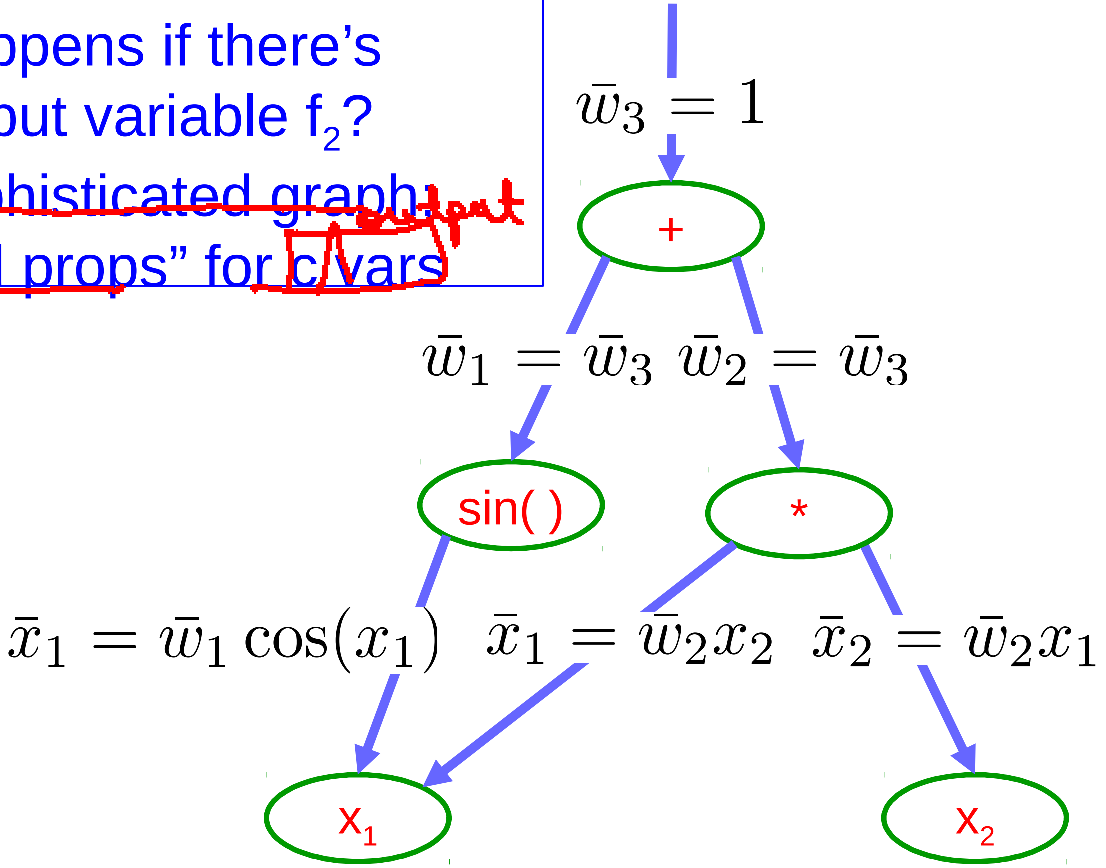
Example: Reverse mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$

Q: What happens if there's another output variable f_2?

A: more sophisticated graph; output

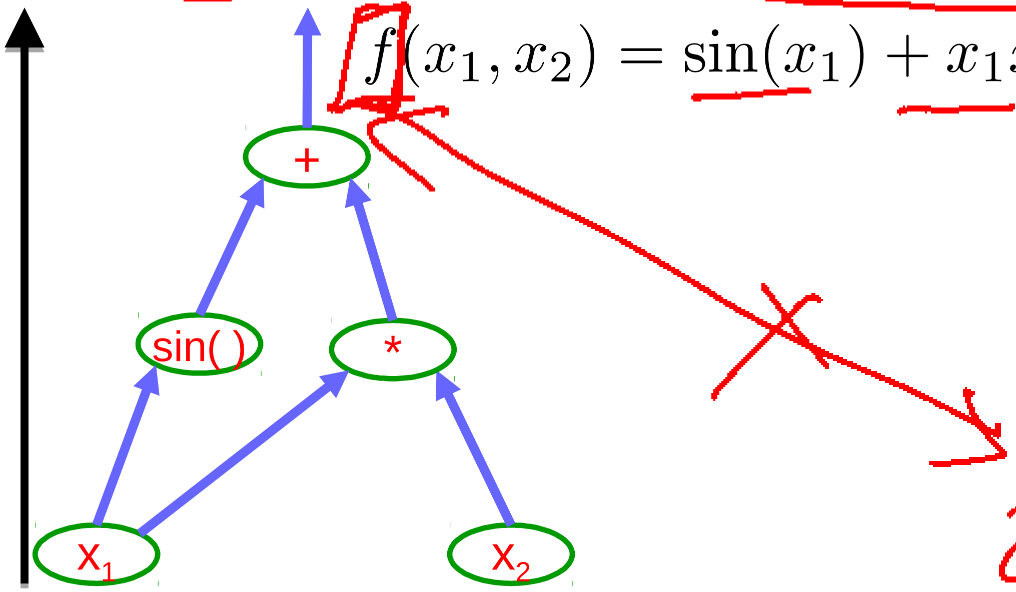
c "backward props" for c/vars

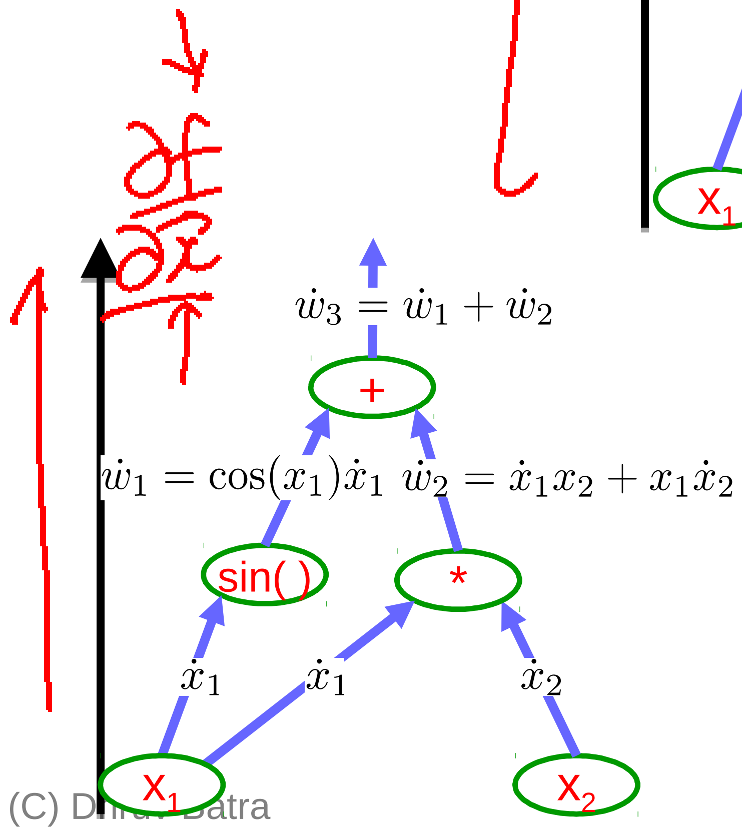


$\bar{w}_3 = 1$
+
$\bar{w}_1 = \bar{w}_3$
$\bar{w}_2 = \bar{w}_3$
sin( )
*
$\bar{x}_1 = \bar{w}_1 \cos(x_1)$
$\bar{x}_1 = \bar{w}_2 x_2$
$\bar{x}_2 = \bar{w}_2 x_1$
x_1
x_2

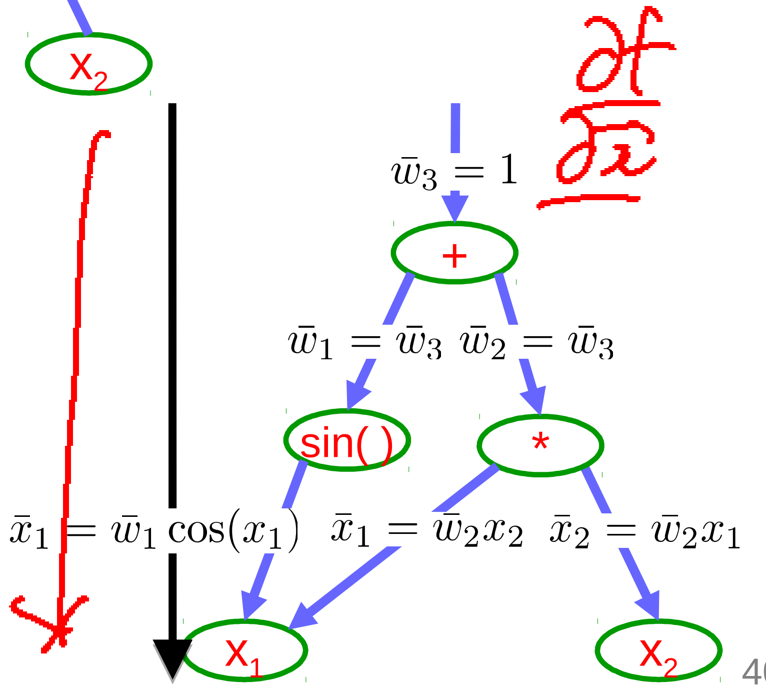
Forward Pass vs Forward mode AD vs Reverse Mode AD

$$f(x_1, x_2) = \sin(x_1) + x_1 x_2$$





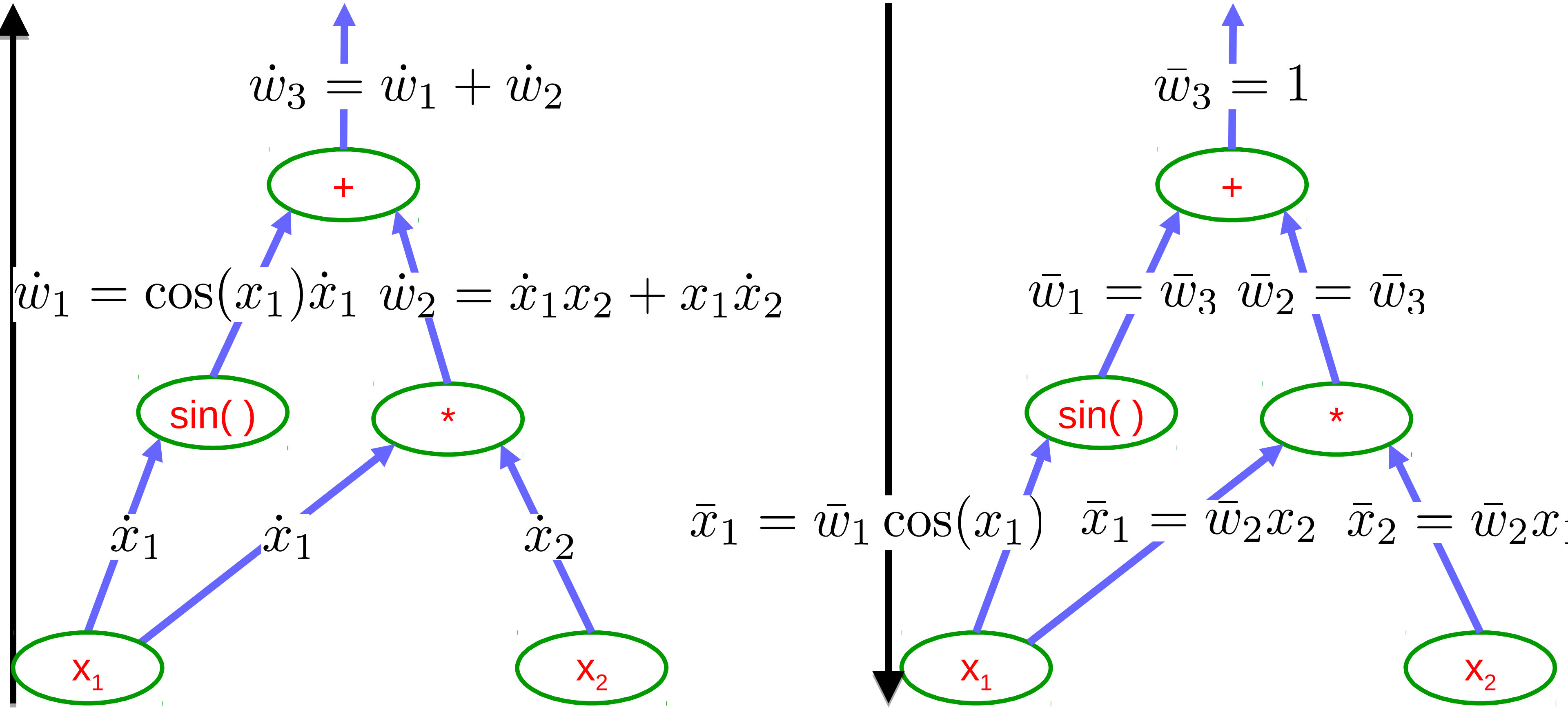
$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
$\frac{\partial f}{\partial x}$
+
sin( )
*
X1
X2



$\bar{w}_3 = 1$
$\bar{w}_1 = \bar{w}_3$
$\bar{w}_2 = \bar{w}_3$
$\bar{x}_1 = \bar{w}_1 \cos(x_1)$
$\bar{x}_1 = \bar{w}_2 x_2$
$\bar{x}_2 = \bar{w}_2 x_1$
$\frac{\partial f}{\partial x}$
+
sin( )
*
X1
X2

Forward mode vs Reverse Mode

- What are the differences?



$\dot{w}_3 = \dot{w}_1 + \dot{w}_2$
$\dot{w}_1 = \cos(x_1)\dot{x}_1$
$\dot{w}_2 = \dot{x}_1 x_2 + x_1 \dot{x}_2$
+
sin( )
*
$\dot{x}_1$
$\dot{x}_1$
$\dot{x}_2$
x_1
x_2
$\bar{w}_3 = 1$
$\bar{w}_1 = \bar{w}_3$
$\bar{w}_2 = \bar{w}_3$
$\bar{x}_1 = \bar{w}_1 \cos(x_1)$
$\bar{x}_1 = \bar{w}_2 x_2$
$\bar{x}_2 = \bar{w}_2 x_1$

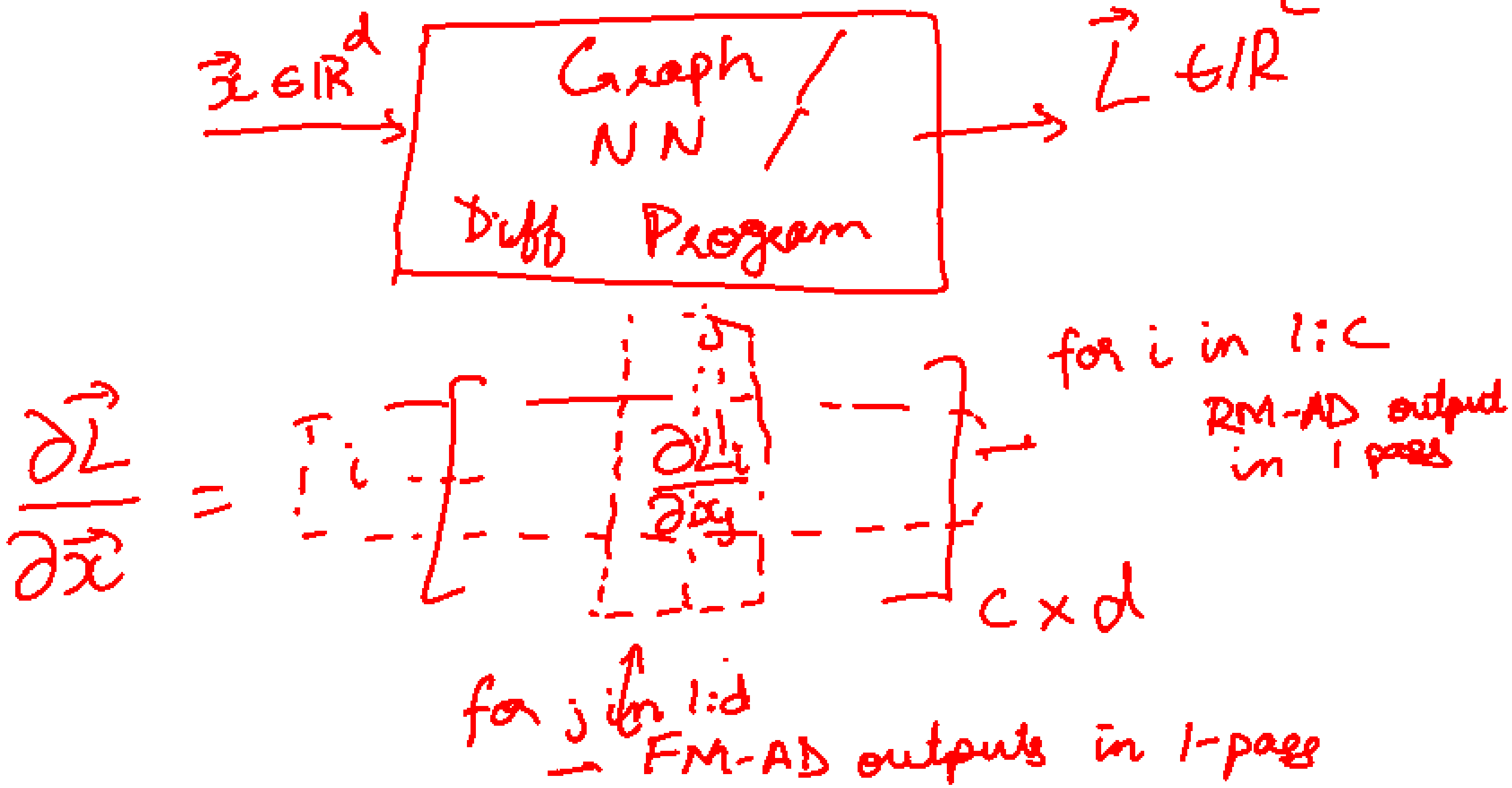
Forward mode vs Reverse Mode

- What are the differences?

- Which one is faster to compute?
  - Forward or backward? Depends Is c>d or d>c?

Forward mode vs Reverse Mode

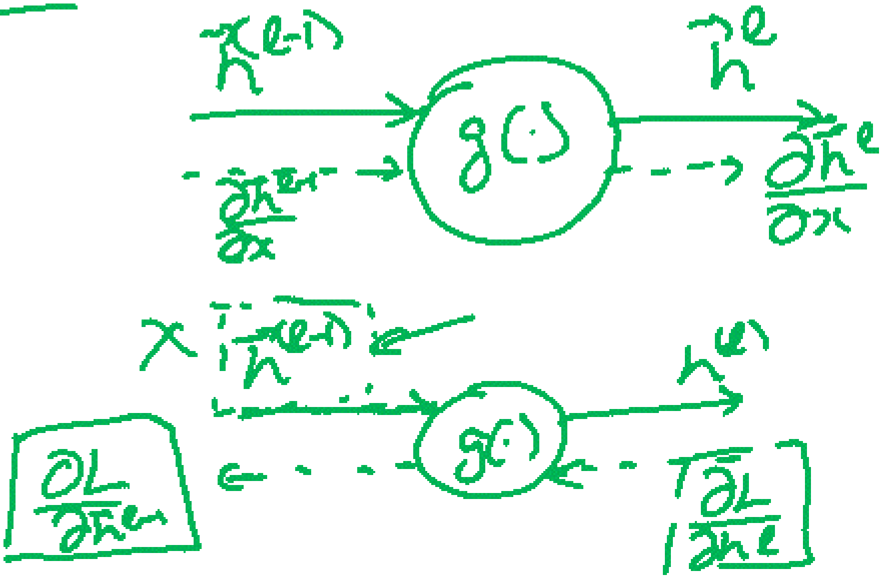
- x ✉ Graph ✉ L
- Intuition of Jacobian



$\vec{x} \in \mathbb{R}^d$
Graph / NN / Diff Program
$\vec{L} \in \mathbb{R}^c$
$\frac{\partial \vec{L}}{\partial \vec{x}} =$
i
j
$\frac{\partial L_i}{\partial x_j}$
c x d
for i in 1:c
RM-AD output in 1 pass
for j in 1:d
FM-AD outputs in 1-pass

Forward mode vs Reverse Mode

- What are the differences?
- Which one is faster to compute?
  - Forward or backward?
- Which one is more memory efficient (less storage)?
  - Forward or backward?



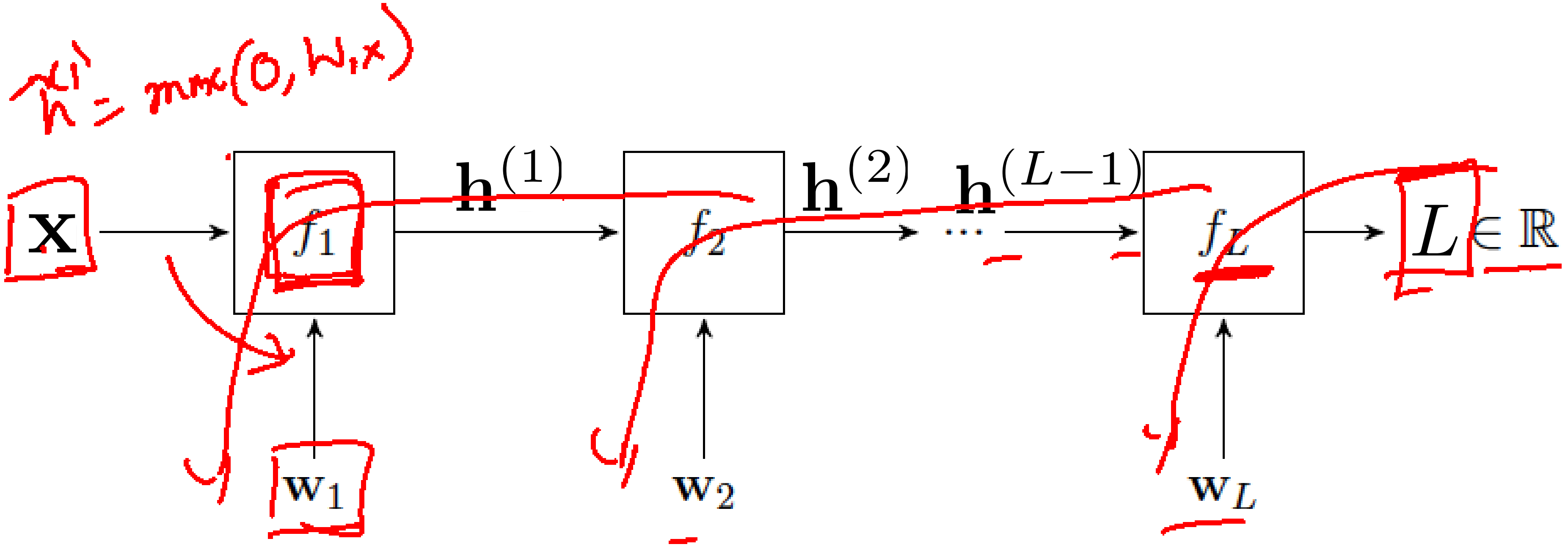
$\tilde{h}^{(l-1)}$
$\tilde{h}^{l}$
$g(\cdot)$
$\frac{\partial h^{l-1}}{\partial x}$
$\frac{\partial h^{l}}{\partial x}$
x
$h^{(l-1)}$
$h^{(l)}$
$g(\cdot)$
$\frac{\partial L}{\partial h^{l-1}}$
$\frac{\partial L}{\partial h^{l}}$

Plan for Today

- Automatic Differentiation
  - (Finish) Forward mode vs Reverse mode AD
  - Backprop
  - Patterns in backprop

Neural Network Computation Graph

$h^{(1)} = \max(0, W_1 x)$



$$\frac{\partial L}{\partial w_{(i)}}$$

Figure Credit: Andrea Vedaldi

Backprop

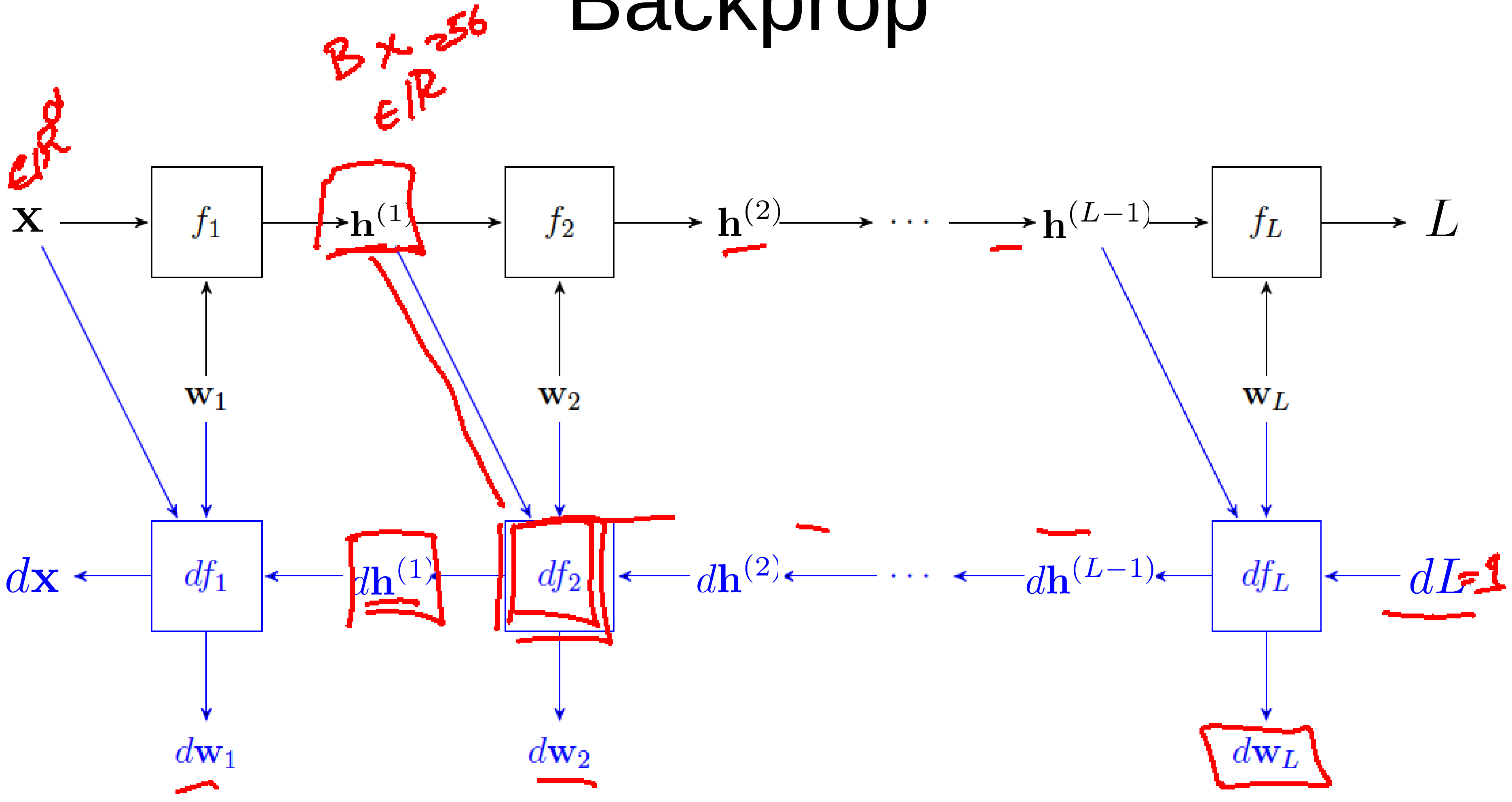
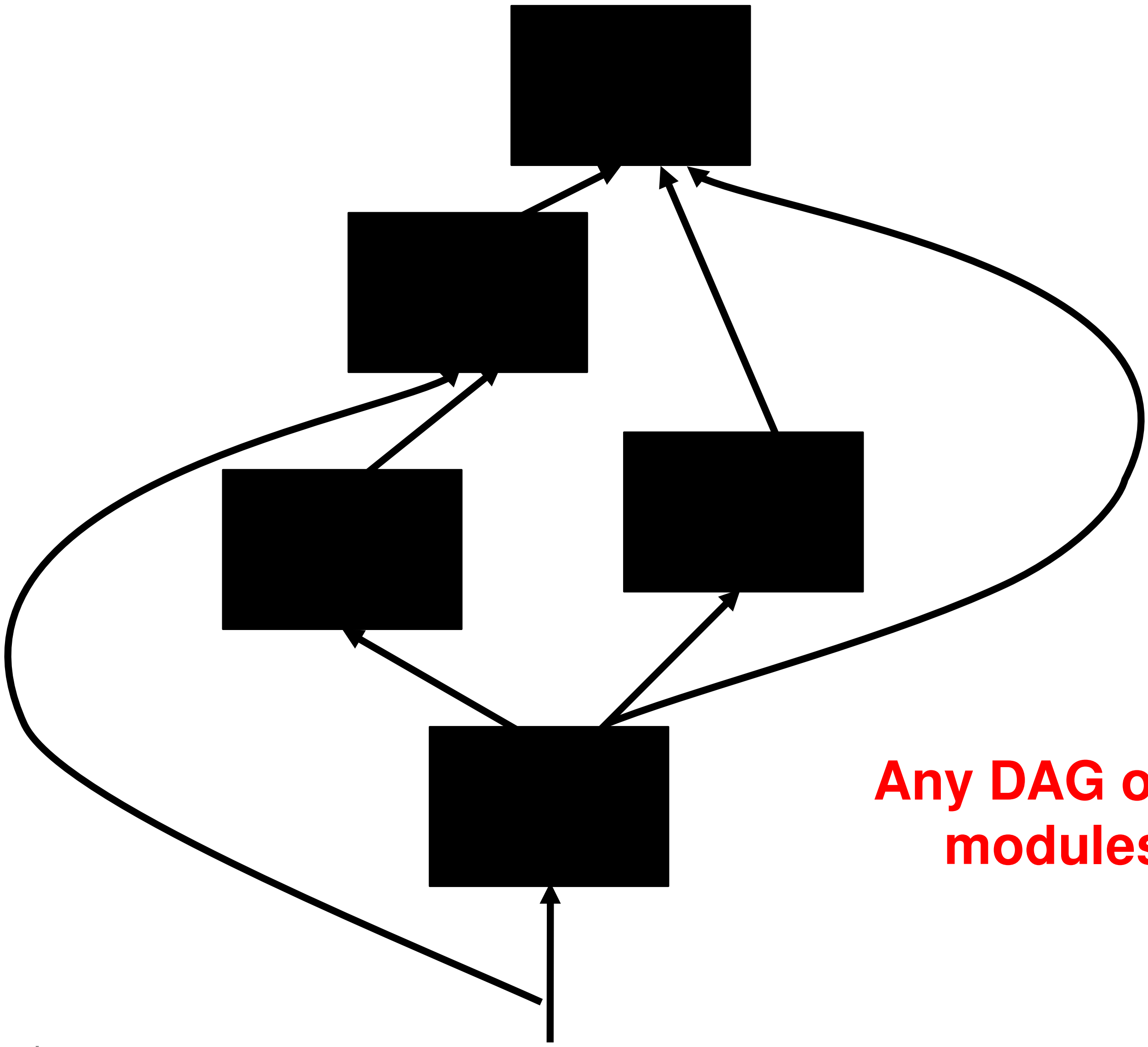


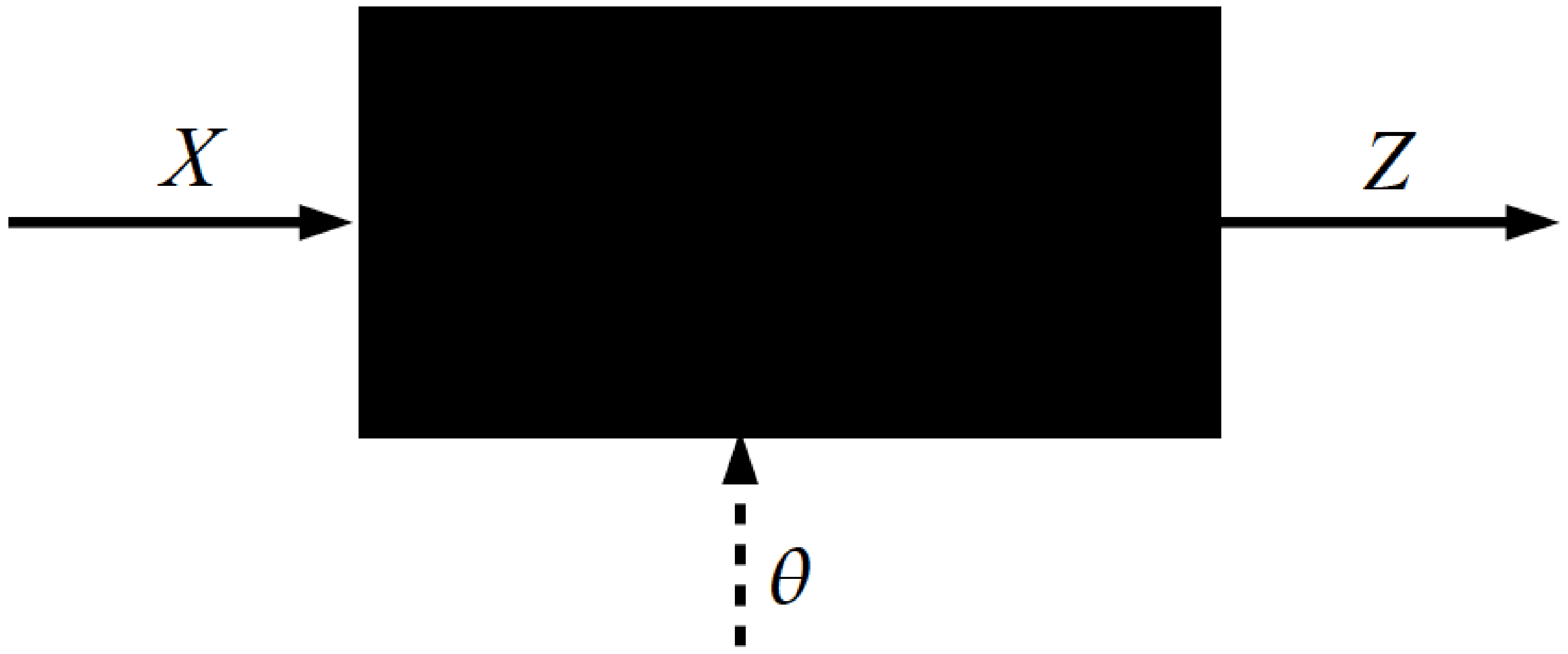
Figure Credit: Andrea Vedaldi

Computational Graph

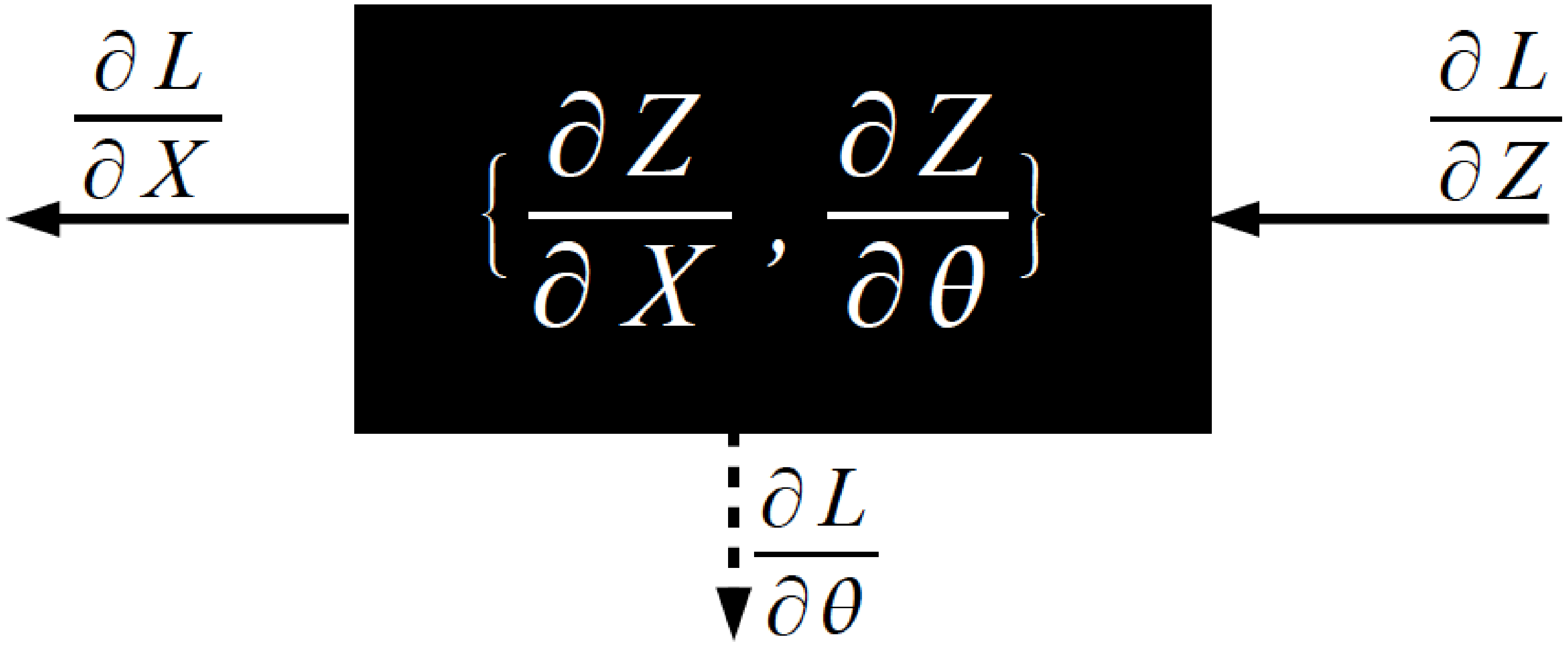


Any DAG of differentiable modules is allowed!

Key Computation: Forward-Prop



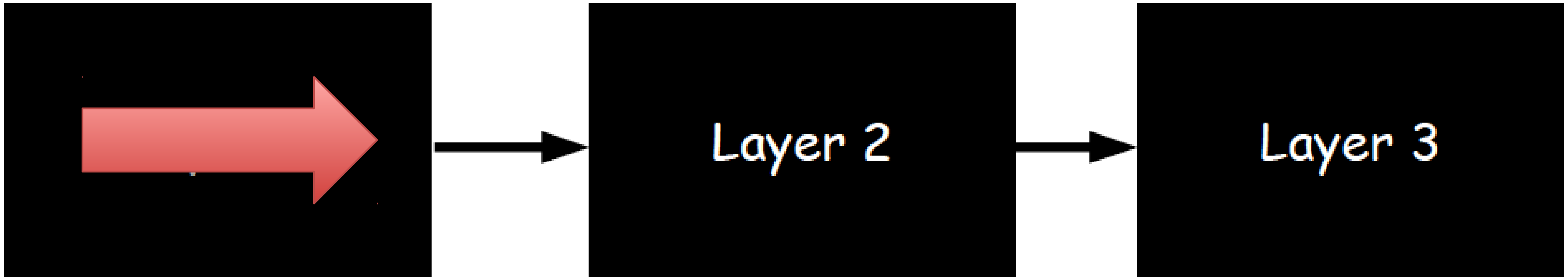
Key Computation: Back-Prop


$\frac{\partial L}{\partial X}$
$\{\frac{\partial Z}{\partial X}, \frac{\partial Z}{\partial \theta}\}$
$\frac{\partial L}{\partial Z}$
$\frac{\partial L}{\partial \theta}$

Slide Credit: Marc'Aurelio Ranzato, Yann LeCun

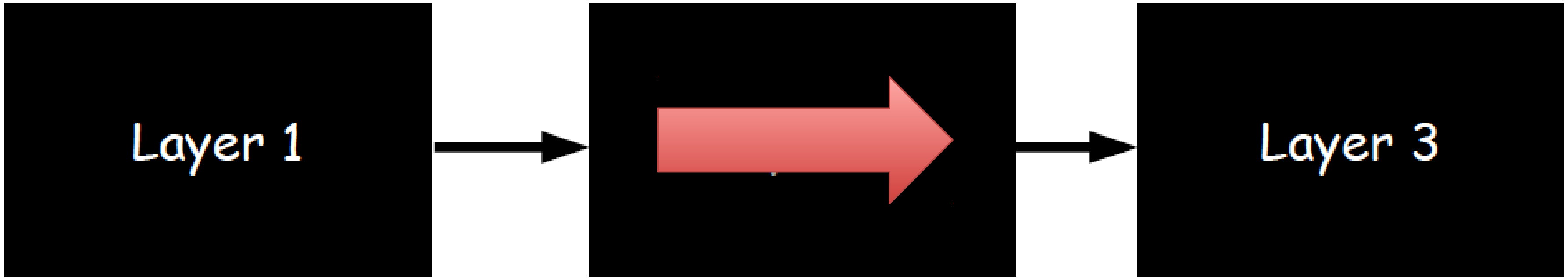
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]



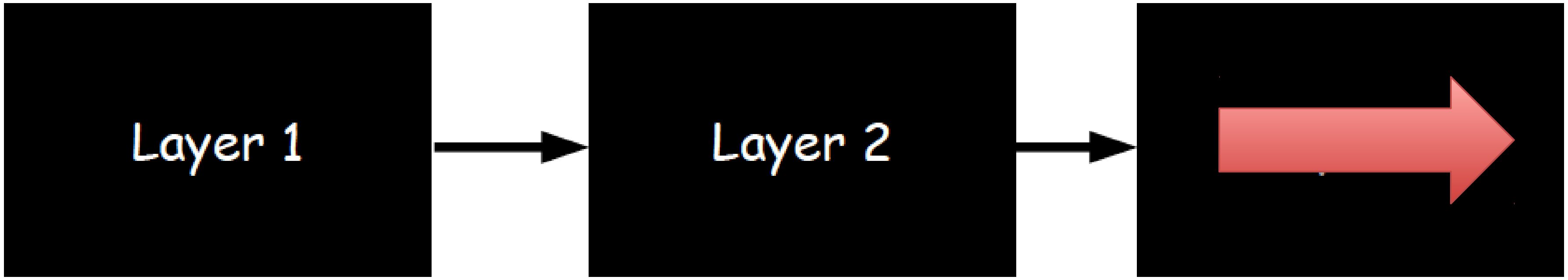
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]



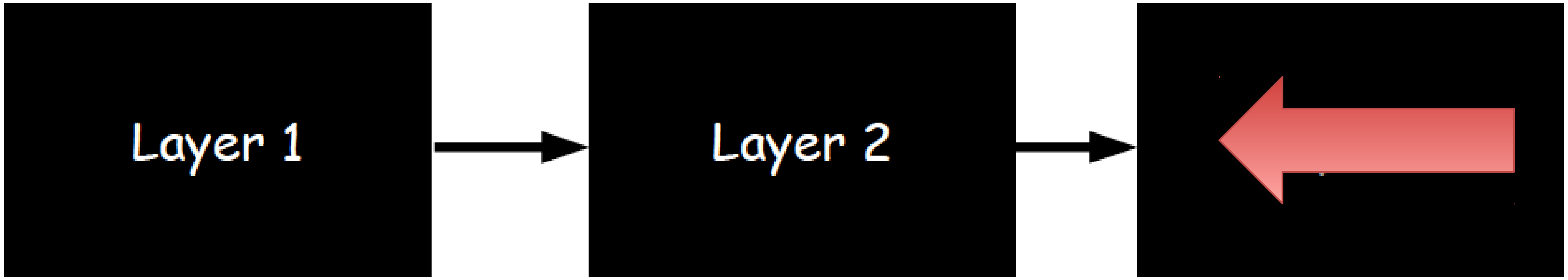
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]



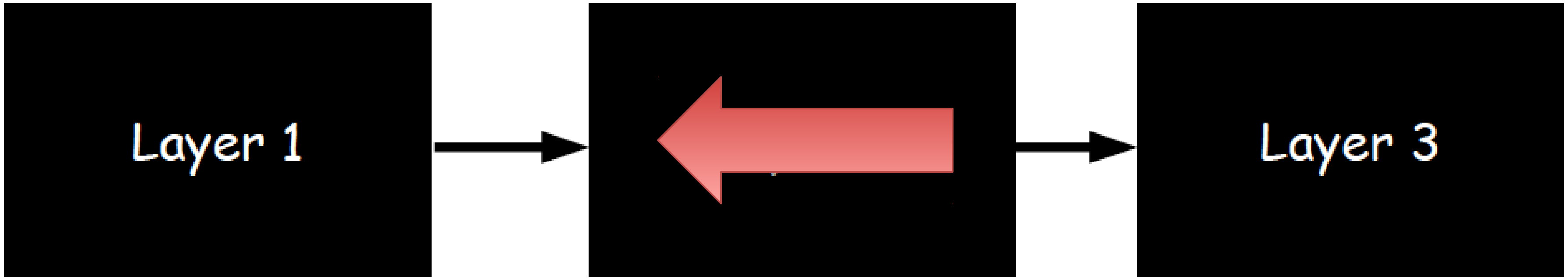
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]



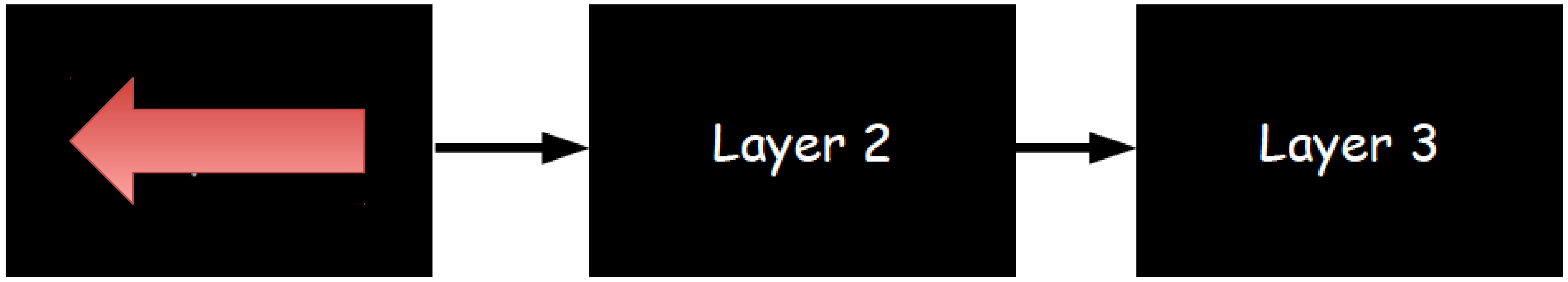
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]



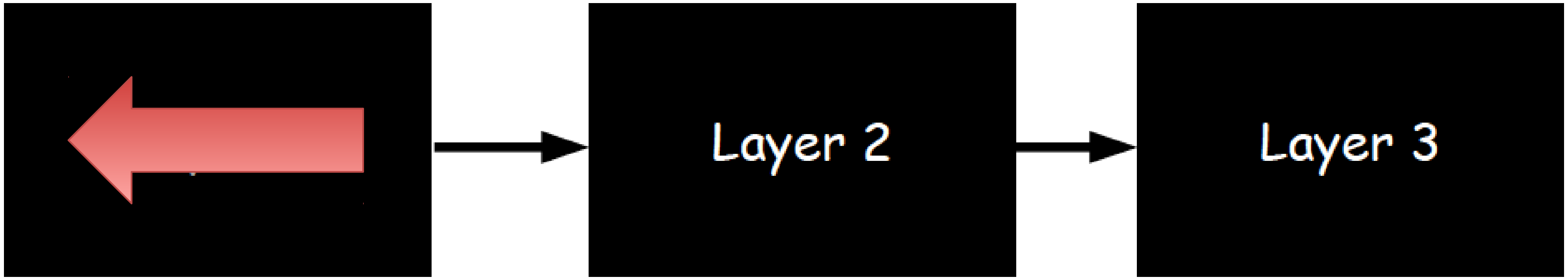
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]



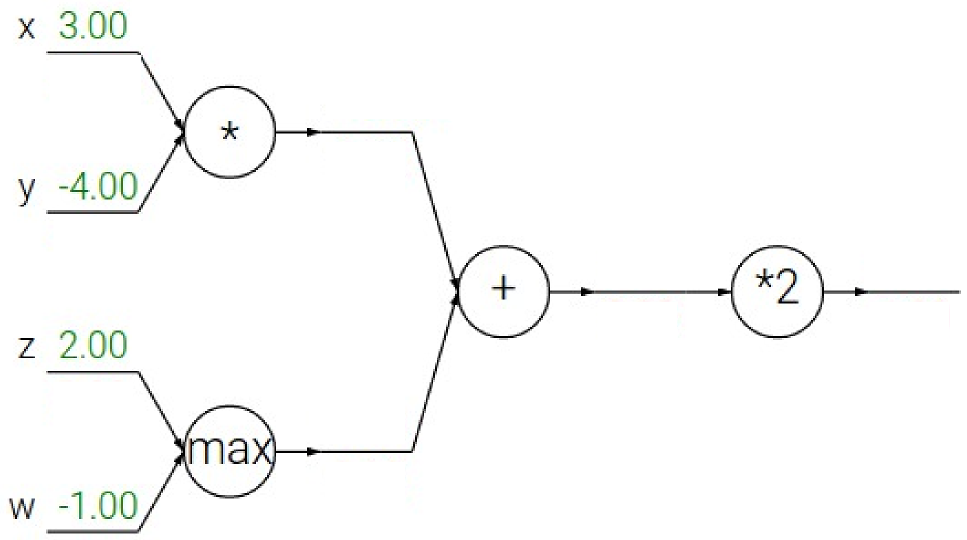
Neural Network Training

- Step 1: Compute Loss on mini-batch [F-Pass]
- Step 2: Compute gradients wrt parameters [B-Pass]
- Step 3: Use gradient to update parameters

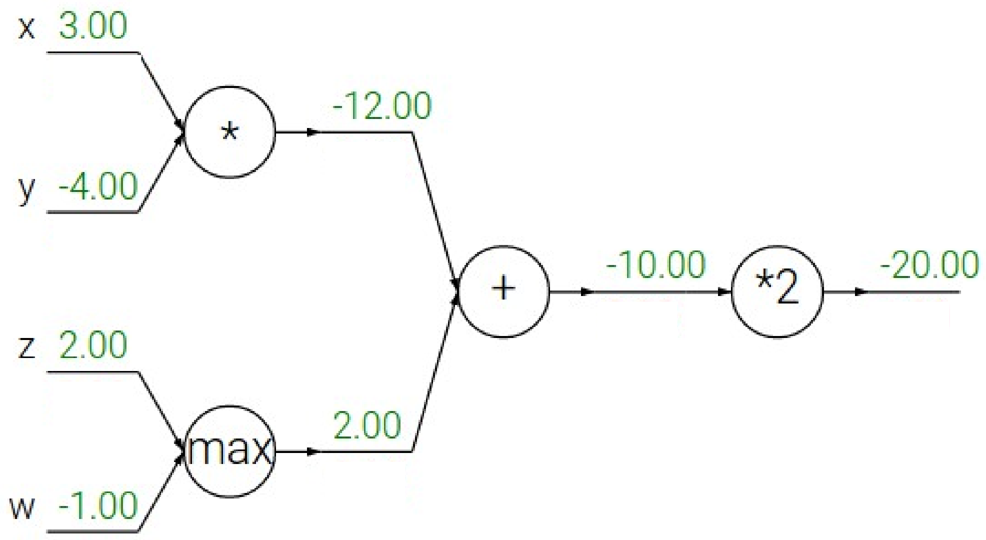


$$\theta \leftarrow \theta - \eta \frac{dL}{d\theta}$$

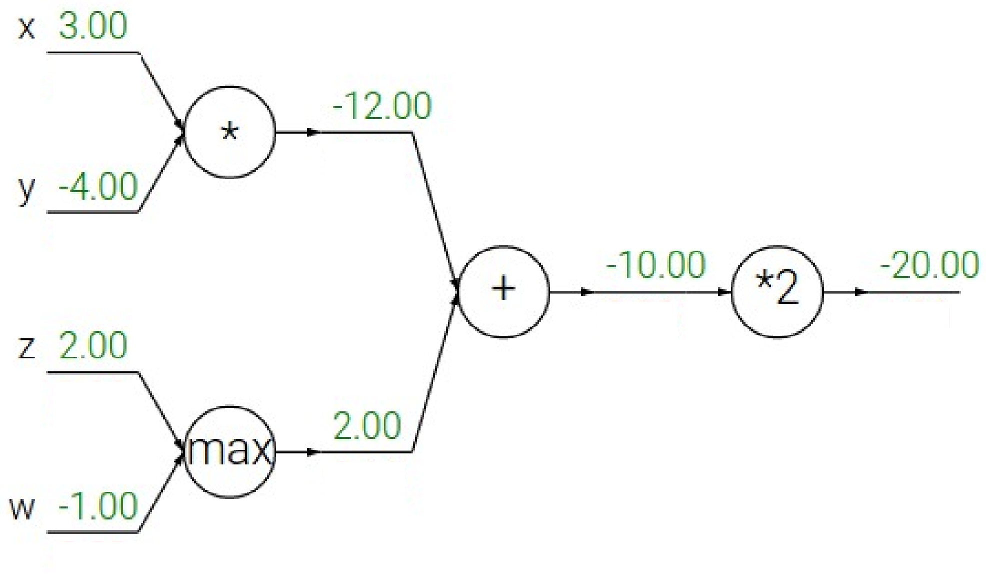
Backpropagation: a simple example



Backpropagation: a simple example

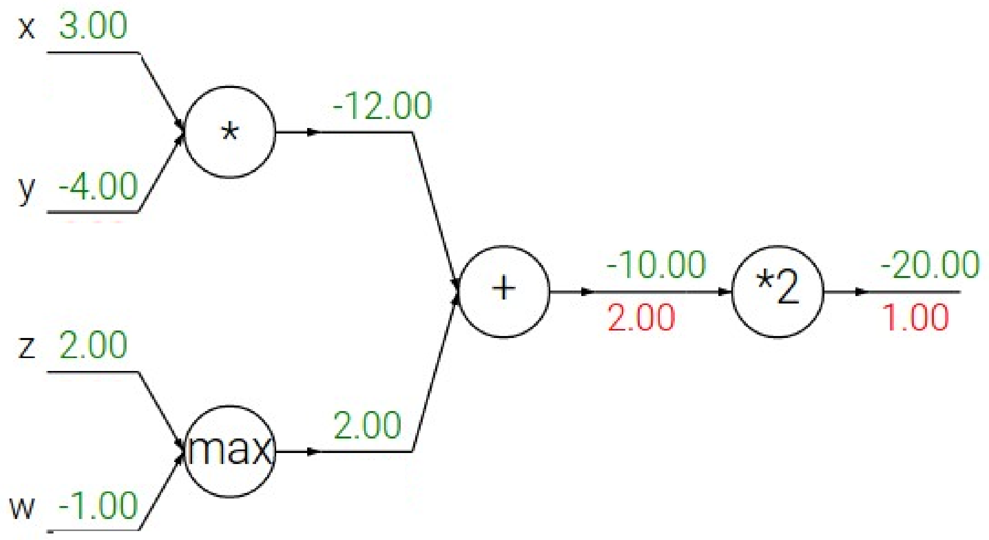


Patterns in backward flow



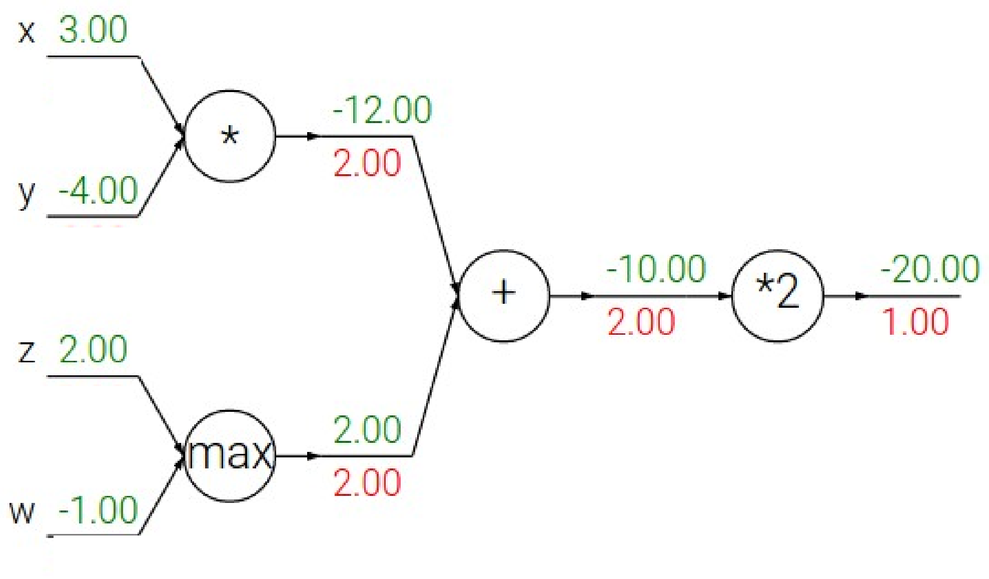
Patterns in backward flow

Q: What is an **add** gate?



Patterns in backward flow

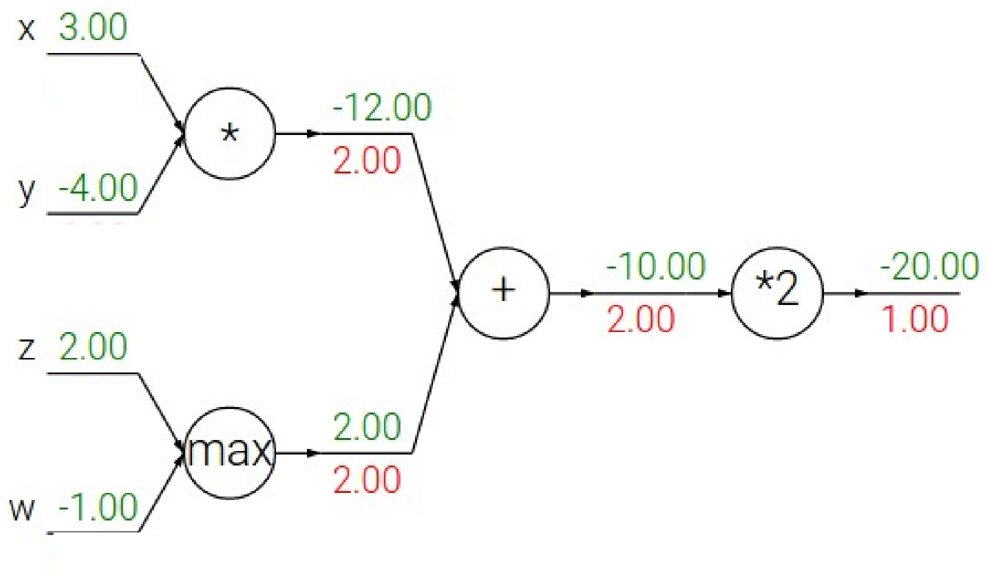
add gate: gradient distributor



Patterns in backward flow

add gate: gradient distributor

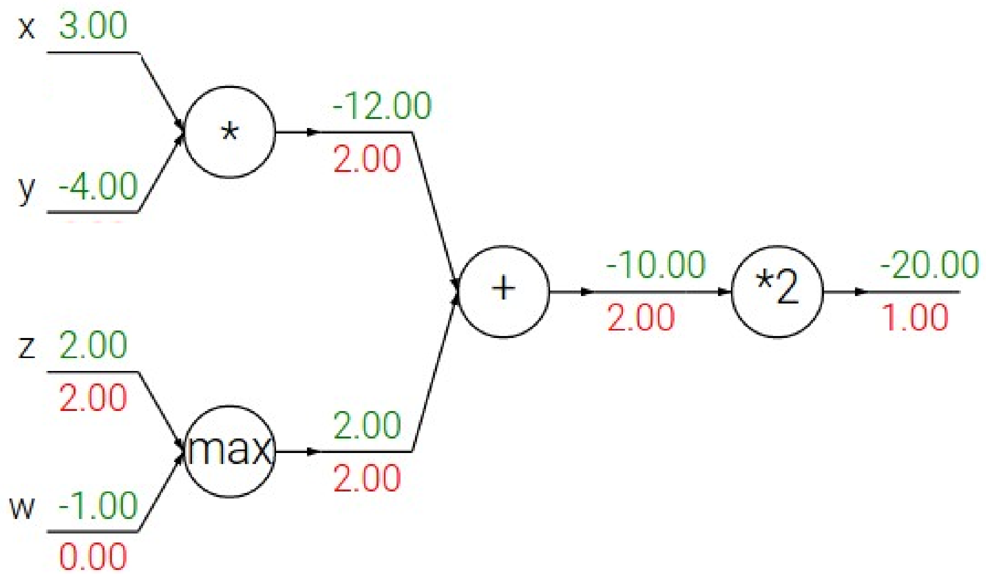
Q: What is a **max** gate?



Patterns in backward flow

add gate: gradient distributor

max gate: gradient router

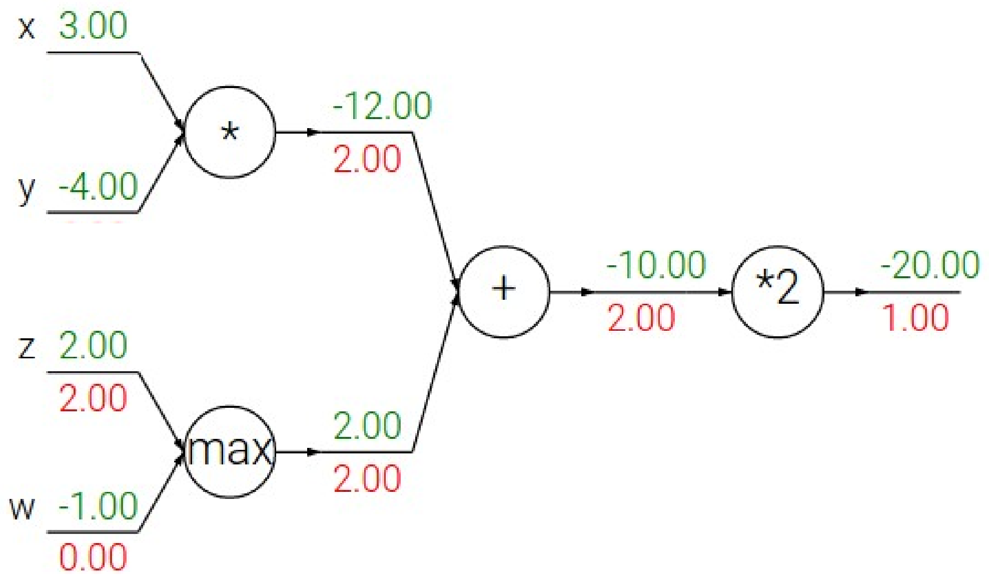


Patterns in backward flow

add gate: gradient distributor

max gate: gradient router

Q: What is a **mul** gate?

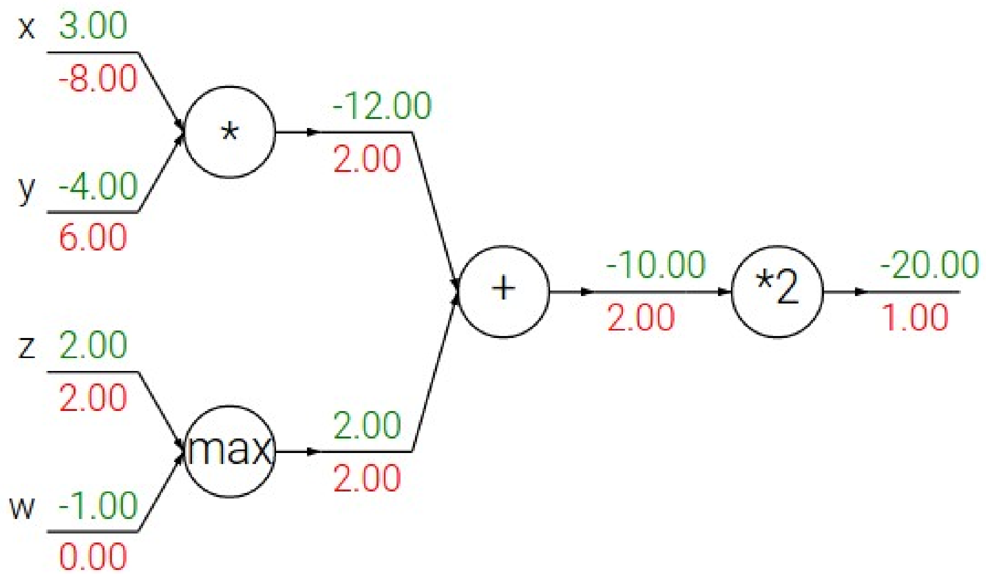


Patterns in backward flow

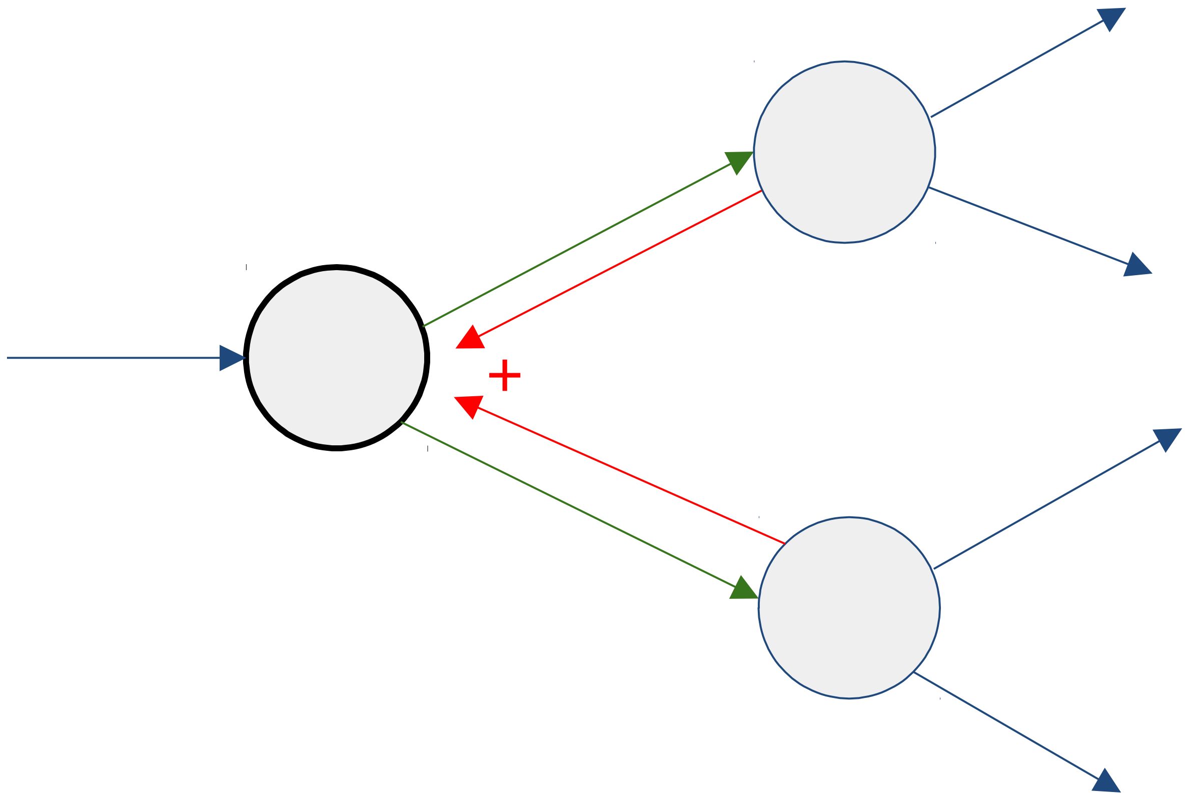
add gate: gradient distributor

max gate: gradient router

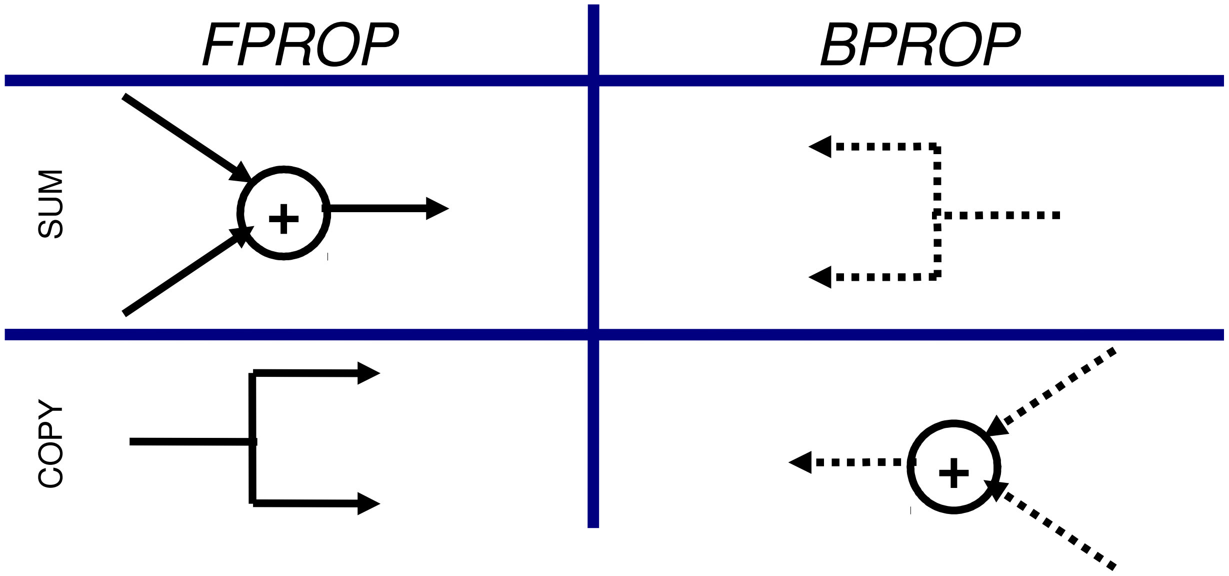
mul gate: gradient switcher



Gradients add at branches



Duality in Fprop and Bprop


FPROP
BPROP
SUM
COPY
+
+