

# 10-301/601: Introduction to Machine Learning

## Lecture 13 – Differentiation

Henry Chai

6/11/24

# Front Matter

- Announcements
  - HW4 released 6/4, due 6/11 (today!) at 11:59 PM
  - HW5 released 6/11 (today!), due 6/18 at 11:59 PM
  - Midterm on Friday, 6/21 from TBD to TBD in TBD
- Recommended Readings
  - None

# Recall: Forward Propagation for Making Predictions

- Input: weights  $W^{(1)}, \dots, W^{(L)}$  and a query data point  $\mathbf{x}$
- Initialize  $\mathbf{o}^{(0)} = [1, \mathbf{x}]^T$
- For  $l = 1, \dots, L$ 
  - $\mathbf{s}^{(l)} = W^{(l)} \mathbf{o}^{(l-1)}$
  - $\mathbf{o}^{(l)} = [1, f(\mathbf{s}^{(l)})]^T$
- Output:  $h_{W^{(1)}, \dots, W^{(L)}}(\mathbf{x}) = \mathbf{o}^{(L)}$

# Recall: Gradient Descent for Learning

- Input:  $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N, \eta^{(0)}$
- Initialize all weights  $W_{(0)}^{(1)}, \dots, W_{(0)}^{(L)}$  to small, random numbers and set  $t = 0$  (???)
- While TERMINATION CRITERION is not satisfied (???)
  - For  $l = 1, \dots, L$ 
    - Compute  $G^{(l)} = \nabla_{W^{(l)}} \ell_{\mathcal{D}} \left( W_{(t)}^{(1)}, \dots, W_{(t)}^{(L)} \right)$  (???)
    - Update  $W^{(l)}$ :  $W_{(t+1)}^{(l)} = W_{(t)}^{(l)} - \eta_0 G^{(l)}$
  - Increment  $t$ :  $t = t + 1$
- Output:  $W_{(t)}^{(1)}, \dots, W_{(t)}^{(L)}$

# Matrix Calculus

		Numerator		
		scalar	vector	matrix
Denominator	Types of Derivatives			
	scalar	$\frac{\partial y}{\partial x}$	$\frac{\partial \mathbf{y}}{\partial x}$	$\frac{\partial \mathbf{Y}}{\partial x}$
	vector	$\frac{\partial y}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{Y}}{\partial \mathbf{x}}$
matrix	$\frac{\partial y}{\partial \mathbf{X}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{X}}$	$\frac{\partial \mathbf{Y}}{\partial \mathbf{X}}$	

# Matrix Calculus: Denominator Layout

- Derivatives of a scalar always have the *same shape* as the entity that the derivative is being taken with respect to.

<i>Types of Derivatives</i>	scalar
<b>scalar</b>	$\frac{\partial y}{\partial x} = \left[ \frac{\partial y}{\partial x} \right]$
<b>vector</b>	$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_P} \end{bmatrix}$
<b>matrix</b>	$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial X_{11}} & \frac{\partial y}{\partial X_{12}} & \cdots & \frac{\partial y}{\partial X_{1Q}} \\ \frac{\partial y}{\partial X_{21}} & \frac{\partial y}{\partial X_{22}} & \cdots & \frac{\partial y}{\partial X_{2Q}} \\ \vdots & & & \vdots \\ \frac{\partial y}{\partial X_{P1}} & \frac{\partial y}{\partial X_{P2}} & \cdots & \frac{\partial y}{\partial X_{PQ}} \end{bmatrix}$

# Matrix Calculus: Denominator Layout

<i>Types of Derivatives</i>	scalar	vector
scalar	$\frac{\partial y}{\partial x} = \left[ \frac{\partial y}{\partial x} \right]$	$\frac{\partial \mathbf{y}}{\partial x} = \left[ \frac{\partial y_1}{\partial x} \quad \frac{\partial y_2}{\partial x} \quad \dots \quad \frac{\partial y_N}{\partial x} \right]$
vector	$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_P} \end{bmatrix}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \dots & \frac{\partial y_N}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_N}{\partial x_2} \\ \vdots & \vdots & \dots & \vdots \\ \frac{\partial y_1}{\partial x_P} & \frac{\partial y_2}{\partial x_P} & \dots & \frac{\partial y_N}{\partial x_P} \end{bmatrix}$

# Three Approaches to Differentiation

• Given  $f: \mathbb{R}^D \rightarrow \mathbb{R}$ , compute  $\nabla_{\mathbf{x}} f(\mathbf{x}) = \partial f(\mathbf{x}) / \partial \mathbf{x}$

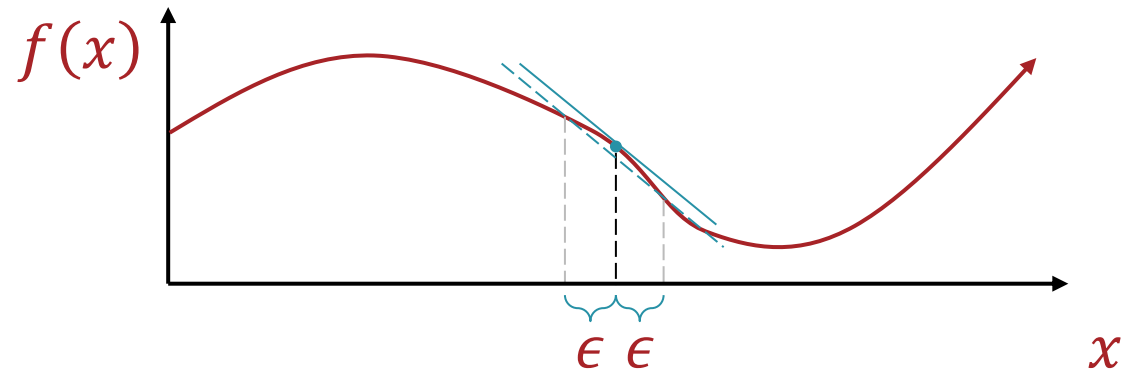
1. Finite difference method
2. Symbolic differentiation
3. Automatic differentiation (reverse mode)



# Approach 1: Finite Difference Method

- Given  $f: \mathbb{R}^D \rightarrow \mathbb{R}$ , compute  $\nabla_x f(\mathbf{x}) = \partial f(\mathbf{x}) / \partial \mathbf{x}$   
$$\frac{\partial f(\mathbf{x})}{\partial x_i} \approx \frac{f(\mathbf{x} + \epsilon \mathbf{d}_i) - f(\mathbf{x} - \epsilon \mathbf{d}_i)}{2\epsilon}$$

where  $\mathbf{d}_i$  is a one-hot vector with a 1 in the  $i^{\text{th}}$  position



- We want  $\epsilon$  to be small to get a good approximation but we run into floating point issues when  $\epsilon$  is too small
- Getting the full gradient requires computing the above approximation for each dimension of the input

# Approach 1: Finite Difference Method Example

- Given

$$y = f(x, z) = e^{xz} + \frac{xz}{\ln(x)} + \frac{\sin(\ln(x))}{xz}$$

what are  $\frac{\partial y}{\partial x}$  and  $\frac{\partial y}{\partial z}$  at  $x = 2, z = 3$ ?

```
>>> import math
>>> y = lambda x,z:
math.exp(x*z)+(x*z)/math.log(x)+math.sin(math.log(x))/(x*z)
>>> x = 2
>>> z = 3
>>> e = 10**-8
>>> dydx = (y(x+e,z)-y(x-e,z))/(2*e)
>>> dydz = (y(x,z+e)-y(x,z-e))/(2*e)
>>> print(dydx, dydz)
```

# Three Approaches to Differentiation

- Given  $f: \mathbb{R}^D \rightarrow \mathbb{R}$ , compute  $\nabla_{\mathbf{x}} f(\mathbf{x}) = \partial f(\mathbf{x}) / \partial \mathbf{x}$
1. Finite difference method
    - Requires the ability to call  $f(\mathbf{x})$
    - Great for checking accuracy of implementations of more complex differentiation methods
    - Computationally expensive for high-dimensional inputs
  2. Symbolic differentiation
  3. Automatic differentiation (reverse mode)

## Approach 2: Symbolic Differentiation

- Given

$$y = f(x, z) = e^{xz} + \frac{xz}{\ln(x)} + \frac{\sin(\ln(x))}{xz}$$

what are  $\frac{\partial y}{\partial x}$  and  $\frac{\partial y}{\partial z}$  at  $x = 2, z = 3$ ?

- Looks like we're gonna need the chain rule!

# The Chain Rule of Calculus

- If  $y = f(z)$  and  $z = g(x)$  then

the corresponding computation graph is

$$\begin{array}{c} x & z & y \\ \square \longrightarrow \square \longrightarrow \square \end{array} \implies \frac{\partial y}{\partial x} = \frac{\partial y}{\partial z} \frac{\partial z}{\partial x}$$

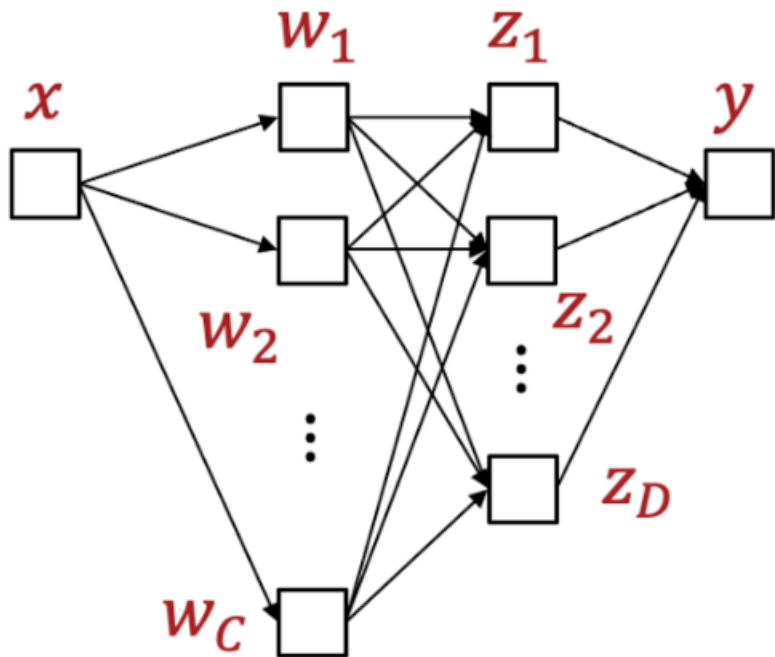
- If  $y = f(z_1, z_2)$  and  $z_1 = g_1(x)$ ,  $z_2 = g_2(x)$  then

$$\begin{array}{c} z_1 & & y \\ x & \swarrow \quad \searrow & \square \\ \square & & \square \\ & \swarrow \quad \searrow & \\ & z_2 & \end{array} \implies \frac{\partial y}{\partial x} = \frac{\partial y}{\partial z_1} \frac{\partial z_1}{\partial x} + \frac{\partial y}{\partial z_2} \frac{\partial z_2}{\partial x}$$

- If  $y = f(\mathbf{z})$  and  $\mathbf{z} = g(x)$  then

$$\begin{array}{c} z_1 & & y \\ x & \swarrow \quad \searrow & \square \\ \square & & \square \\ & \swarrow \quad \searrow & \\ & \vdots & \\ & z_D & \end{array} \implies \frac{\partial y}{\partial x} = \sum_{d=1}^D \frac{\partial y}{\partial z_d} \frac{\partial z_d}{\partial x}$$

Suppose  $y = f(\mathbf{z})$ ,  $\mathbf{z} = h(\mathbf{w})$  and  $\mathbf{w} = g(x)$ . Does the equation  $\frac{\partial y}{\partial x} = \sum_{d=1}^D \frac{\partial y}{\partial z_d} \frac{\partial z_d}{\partial x}$  still hold?



Yes

0%

No

0%

Unsure

0%

## Approach 2: Symbolic Differentiation

- Given

$$y = f(x, z) = e^{xz} + \frac{xz}{\ln(x)} + \frac{\sin(\ln(x))}{xz}$$

what are  $\frac{\partial y}{\partial x}$  and  $\frac{\partial y}{\partial z}$  at  $x = 2, z = 3$ ?

$$\begin{aligned}\frac{\partial y}{\partial x} &= \frac{\partial}{\partial x} (e^{xz}) + \frac{\partial}{\partial x} \left( \frac{xz}{\ln(x)} \right) + \frac{\partial}{\partial x} \left( \frac{\sin(\ln(x))}{xz} \right) \\ &= ze^{xz} + \frac{z}{\ln(x)} - \frac{z}{\ln(x)^2} + \frac{\cos(\ln(x))}{x^2 z} - \frac{\sin(\ln(x))}{x^2 z} \\ &= 3e^6 + \frac{3}{\ln(2)} - \frac{3}{\ln(2)^2} + \frac{\cos(\ln(2))}{12} - \frac{\sin(\ln(2))}{12}\end{aligned}$$

$$\begin{aligned}\frac{\partial y}{\partial z} &= \frac{\partial}{\partial z} (e^{xz}) + \frac{\partial}{\partial z} \left( \frac{xz}{\ln(x)} \right) + \frac{\partial}{\partial z} \left( \frac{\sin(\ln(x))}{xz} \right) \\ &= 2e^6 + \frac{2}{\ln(2)} - \frac{\sin(\ln(2))}{18}\end{aligned}$$

# Three Approaches to Differentiation

- Given  $f: \mathbb{R}^D \rightarrow \mathbb{R}$ , compute  $\nabla_{\mathbf{x}} f(\mathbf{x}) = \partial f(\mathbf{x}) / \partial \mathbf{x}$
- 1. Finite difference method
  - Requires the ability to call  $f(\mathbf{x})$
  - Great for checking accuracy of implementations of more complex differentiation methods
  - Computationally expensive for high-dimensional inputs
- 2. Symbolic differentiation
  - Requires systematic knowledge of derivatives
  - Can be computationally expensive if poorly implemented
- 3. Automatic differentiation (reverse mode)



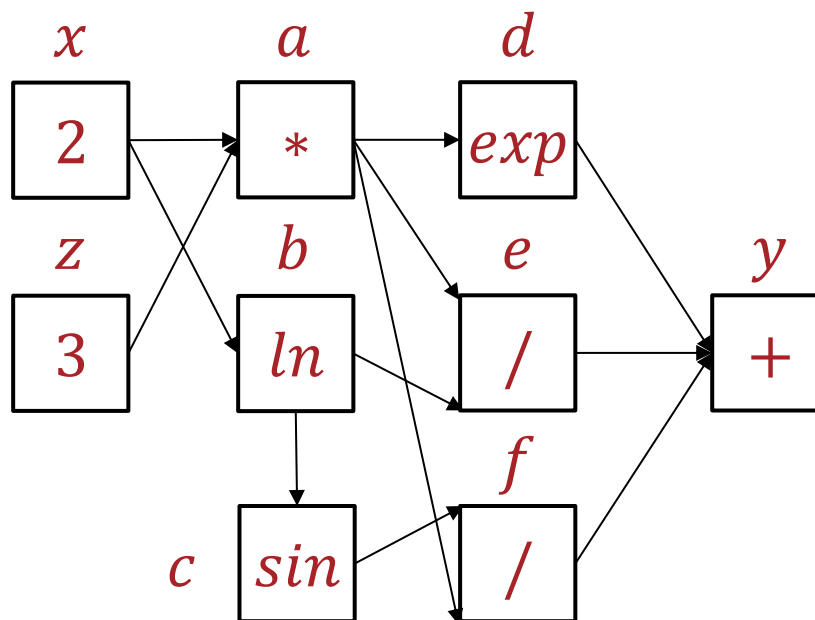
- Given

$$y = f(x, z) = e^{xz} + \frac{xz}{\ln(x)} + \frac{\sin(\ln(x))}{xz}$$

what are  $\frac{\partial y}{\partial x}$  and  $\frac{\partial y}{\partial z}$  at  $x = 2, z = 3$ ?

- First define some intermediate quantities, draw the computation graph and run the “forward” computation

$$\begin{aligned} a &= xz \\ b &= \ln(x) \\ c &= \sin(b) \\ d &= e^a \\ e &= a/b \\ f &= c/a \\ y &= d + e + f \end{aligned}$$



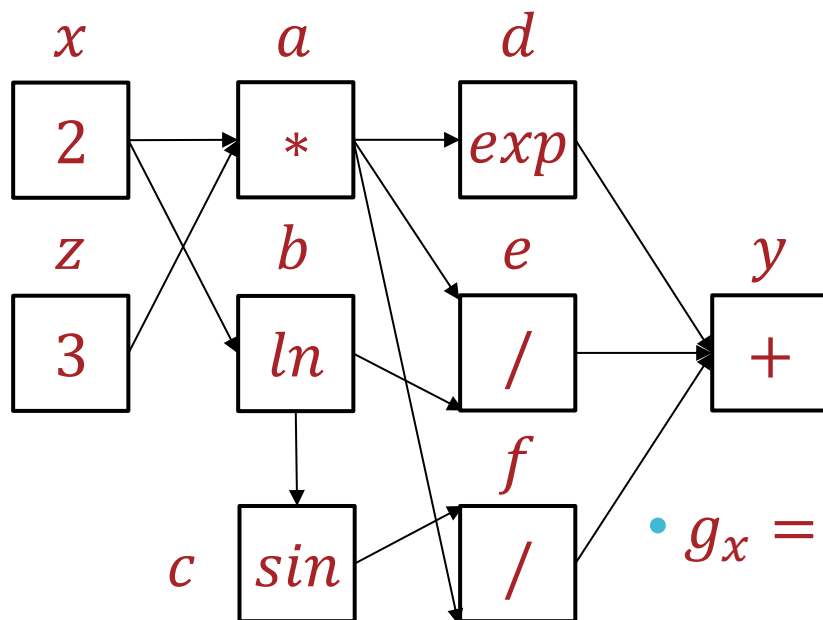
## Approach 3: Automatic Differentiation (reverse mode)

- Given

$$y = f(x, z) = e^{xz} + \frac{xz}{\ln(x)} + \frac{\sin(\ln(x))}{xz}$$

what are  $\frac{\partial y}{\partial x}$  and  $\frac{\partial y}{\partial z}$  at  $x = 2, z = 3$ ?

- Then compute partial derivatives, starting from  $y$  and working back



- $g_y = \frac{\partial y}{\partial y} = 1$

- $g_d = g_e = g_f = 1$

- $g_c = \frac{\partial y}{\partial c} = \frac{\partial y}{\partial f} \frac{\partial f}{\partial c} = g_f \left( \frac{1}{a} \right)$

- $g_b = \frac{\partial y}{\partial b} = \frac{\partial y}{\partial e} \frac{\partial e}{\partial b} + \frac{\partial y}{\partial c} \frac{\partial c}{\partial b}$   
 $= g_e \left( -\frac{a}{b^2} \right) + g_c (\cos(b))$

- $g_a = \frac{\partial y}{\partial a} = \frac{\partial y}{\partial f} \frac{\partial f}{\partial a} + \frac{\partial y}{\partial e} \frac{\partial e}{\partial a} + \frac{\partial y}{\partial d} \frac{\partial d}{\partial a}$   
 $= g_f \left( \frac{-c}{a^2} \right) + g_e \left( \frac{1}{b} \right) + g_d (e^a)$

- $g_x = \frac{\partial y}{\partial x} = \frac{\partial y}{\partial b} \frac{\partial b}{\partial x} + \frac{\partial y}{\partial a} \frac{\partial a}{\partial x} = g_b \left( \frac{1}{x} \right) + g_a (z)$

- $g_z = \frac{\partial y}{\partial z} = \frac{\partial y}{\partial a} \frac{\partial a}{\partial z} = g_a (x)$

## Approach 3: Automatic Differentiation (reverse mode)

# Three Approaches to Differentiation

- Given  $f: \mathbb{R}^D \rightarrow \mathbb{R}$ , compute  $\nabla_{\mathbf{x}} f(\mathbf{x}) = \partial f(\mathbf{x}) / \partial \mathbf{x}$ 
  1. Finite difference method
    - Requires the ability to call  $f(\mathbf{x})$
    - Great for checking accuracy of implementations of more complex differentiation methods
    - Computationally expensive for high-dimensional inputs
  2. Symbolic differentiation
    - Requires systematic knowledge of derivatives
    - Can be computationally expensive if poorly implemented
  3. Automatic differentiation (reverse mode)
    - Requires systematic knowledge of derivatives *and* an algorithm for computing  $f(\mathbf{x})$
    - Computational cost of computing  $\partial f(\mathbf{x}) / \partial \mathbf{x}$  is proportional to the cost of computing  $f(\mathbf{x})$

# Key Takeaways

- Denominator layout for matrix calculus
- Finite difference method is a simple but computationally expensive approximation technique
  - ***You should use this to unit test your implementation of backpropagation!***
- Symbolic differentiation is the “default” differentiation method but can also be computationally expensive
- Automatic differentiation (reverse mode) saves intermediate quantities for computational efficiency
  - Backpropagation is an instance of automatic differentiation in the reverse mode