

# DSCI 565: PRELIMINARIES

*This content is protected and may not  
be shared, uploaded, or distributed.*

Ke-Thia Yao

Lecture 2: 2025 August 27

# Tensor

2

- For machine learning, tensor is just a **n-dimensional array**
- In physics and engineering, tensor has other interpretations
  - For example, in physics for each point a physical object the stress may have to be represent as 3-d matrix (i.e., stress tensor).  
Tensors over multiple points is called tensor field, but people sometimes just call it tensor

$x$   
scalar

0-dim

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

vector

1-dim

$$A = \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{mn} \end{bmatrix}$$

matrix

2-dim

# Tensor

3

- Need to use tensors, because data is high-dimensional
  - An image is three-dimensional height, width and depth (3 colors)
  - A batch of images is four-dimensional
  - A video is four-dimensional height, width and time
  - A batch of videos is five-dimensional
- See notebook: chapter\_preliminaries/ndarray.ipynb

# Data Preparation

4

- Numerical packages, like PyTorch and NumPy, are designed for efficiency in terms of speed and memory usage
- They assume all the tensor elements are numbers, which is not the case for real-world data
- The Pandas library is designed data wrangling
- We will not cover Pandas beyond this Jupyter notebook
  - ▣ chapter\_preliminaries/pandas.ipynb
- A good Pandas book by the creator of Pandas is:
  - ▣ McKinney, Wes. 2022. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*. 3rd edition. O'Reilly Media.

# Linear Algebra

5

- We can define operations over tensors. Two basic operations are:

- Addition

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbf{x} + \mathbf{y} = \begin{bmatrix} x_1 + y_1 \\ \vdots \\ x_n + y_n \end{bmatrix}$$

- Scalar multiplication

$$\alpha \mathbf{x} = \begin{bmatrix} \alpha x_1 \\ \vdots \\ \alpha x_n \end{bmatrix}$$

- Vector Space: Set of objects with defined addition and scalar multiplication and satisfies some minor axioms (such as having element  $\mathbf{0} = \mathbf{x} + (-\mathbf{x})$ )
- n-dimensional tensors are vector spaces

# Other Operations

6

- Element-wise product or Hadamard product

$$\mathbf{x} \odot \mathbf{y} = \begin{bmatrix} x_1 * y_1 \\ \vdots \\ x_n * y_n \end{bmatrix}$$

- Summation

$$\sum_{i=1}^n x_i, \quad \sum_{i=1}^n \sum_{j=1}^m x_{ij}$$

# Dot Product (aka Inner Product)

7

Given two vectors  $x, y \in \mathbb{R}^d$

$$x^T y = \langle x, y \rangle = \sum_{i=1}^d x_i y_i$$

$x^T y$

- Dot product have many uses
  - If  $x$  is a weight vector,  
then the dot product is weighted average of elements of  $y$ .
  - The length of projection of  $x$  onto  $y$  is  $\|x\|\cos(\theta)$ , where  $\cos(\theta) = \frac{x^T y}{\|x\|\|y\|}$

# Matrix Vector

8

 $a_{1 \times 1}$ 

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$\mathbf{b} = Ax \quad b_i = \sum_{j=1}^n a_{ij}x_j \quad ik, k \rightarrow i$$

$$A = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix}$$

$$Ax = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{x} \\ \mathbf{a}_2^\top \mathbf{x} \\ \vdots \\ \mathbf{a}_m^\top \mathbf{x} \end{bmatrix}$$

# Matrix-Matrix Multiple

9

Say we have two matrices  $\mathbf{A} \in \mathbb{R}^{n \times p}$  and  $\mathbf{B} \in \mathbb{R}^{p \times m}$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{bmatrix}_{n \times k} \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{km} \end{bmatrix}_{k \times m}$$

Then  $\mathbf{C} \in \mathbb{R}^{n \times m}$

$$\mathbf{C} = \mathbf{AB}$$

$$c_{ij} = \sum_{k=1}^p a_{ik} b_{kj} \quad ik, kj \rightarrow ij$$

# Matrix-Matrix Multiple

10

Say we have two matrices  $A \in \mathbb{R}^{n \times p}$  and  $B \in \mathbb{R}^{p \times m}$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{np} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p1} & b_{p2} & \cdots & b_{pm} \end{bmatrix}$$

$$C = AB = \begin{bmatrix} \mathbf{a}_1^\top \\ \mathbf{a}_2^\top \\ \vdots \\ \mathbf{a}_n^\top \end{bmatrix} [\mathbf{b}_1 \quad \mathbf{b}_2 \quad \cdots \quad \mathbf{b}_m] = \begin{bmatrix} \mathbf{a}_1^\top \mathbf{b}_1 & \mathbf{a}_1^\top \mathbf{b}_2 & \cdots & \mathbf{a}_1^\top \mathbf{b}_m \\ \mathbf{a}_2^\top \mathbf{b}_1 & \mathbf{a}_2^\top \mathbf{b}_2 & \cdots & \mathbf{a}_2^\top \mathbf{b}_m \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_n^\top \mathbf{b}_1 & \mathbf{a}_n^\top \mathbf{b}_2 & \cdots & \mathbf{a}_n^\top \mathbf{b}_m \end{bmatrix}$$

# Norms

11

- Norms measurement the magnitude of a vector

- $l_2$  norm

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$$

- $l_1$  norm

$$\|x\|_1 = \sum_{i=1}^n abs(x_i)$$

- $l_p$  norm

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}$$

# Norms

12

## □ Frobenius Norm for matrices

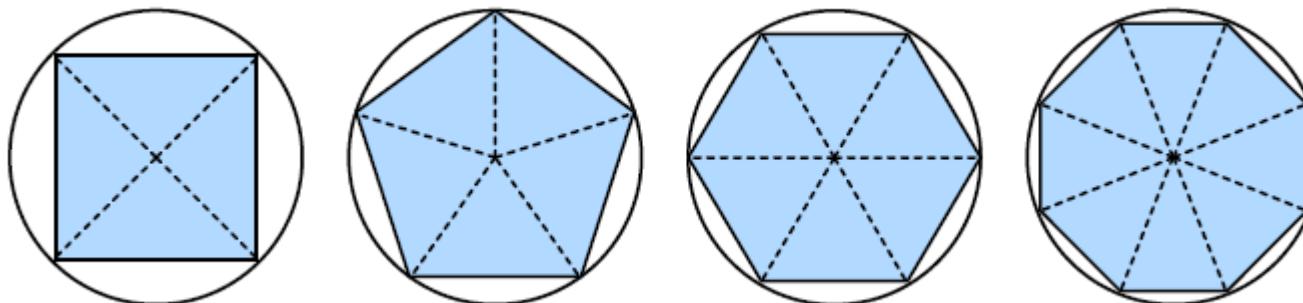
$$\|X\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n x_{ij}^2}$$

- See Notebook: chapter\_preliminaries/linear-algebra.ipynb

# Calculus

14

- Idea of limits: approximate area of a circle by  $n$  triangles

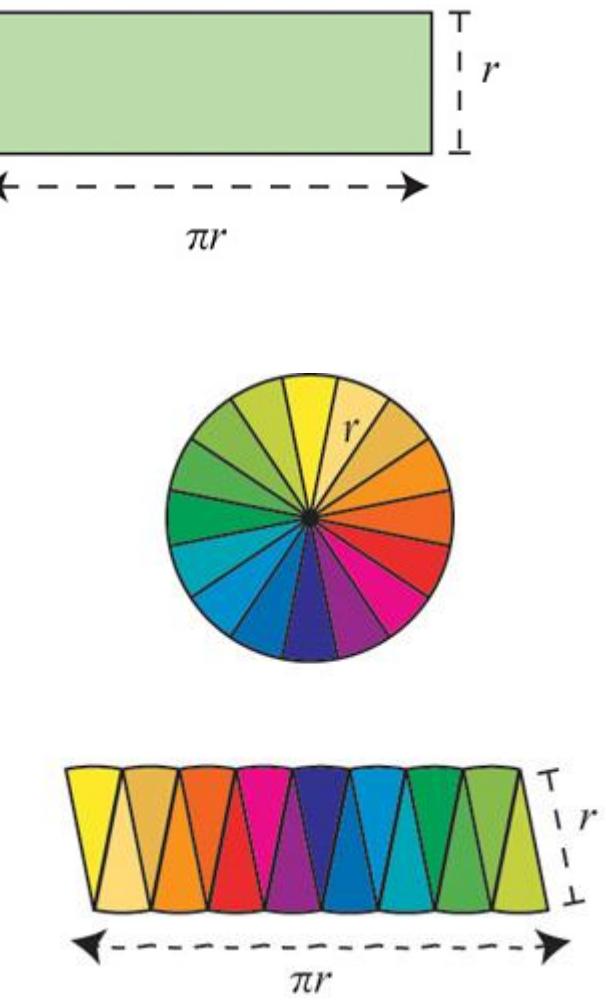
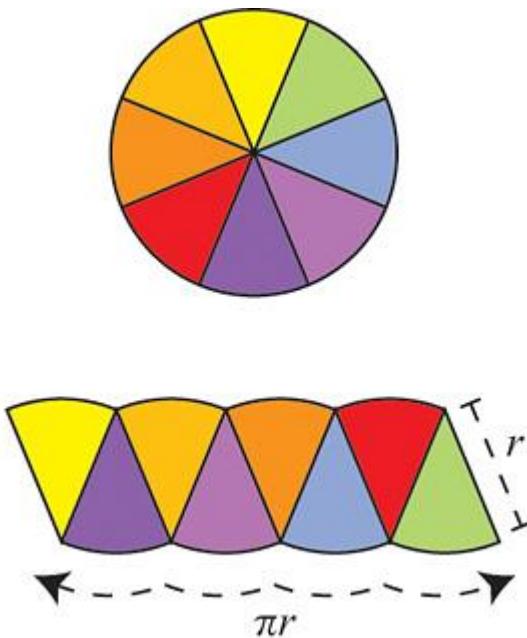
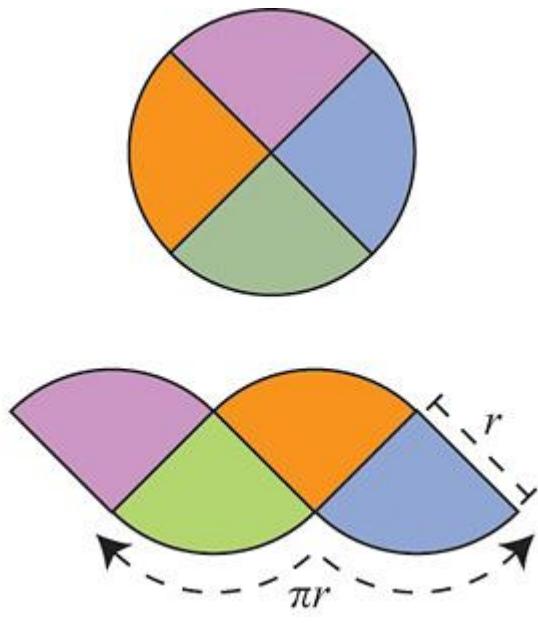


$$n \times \frac{r}{2} \times \frac{2\pi r}{n}$$

# Calculus

15

- Approximate circle area by a rectangle



<https://archive.nytimes.com/opinionator.blogs.nytimes.com/2010/04/04/take-it-to-the-limit/>

# Derivatives

16

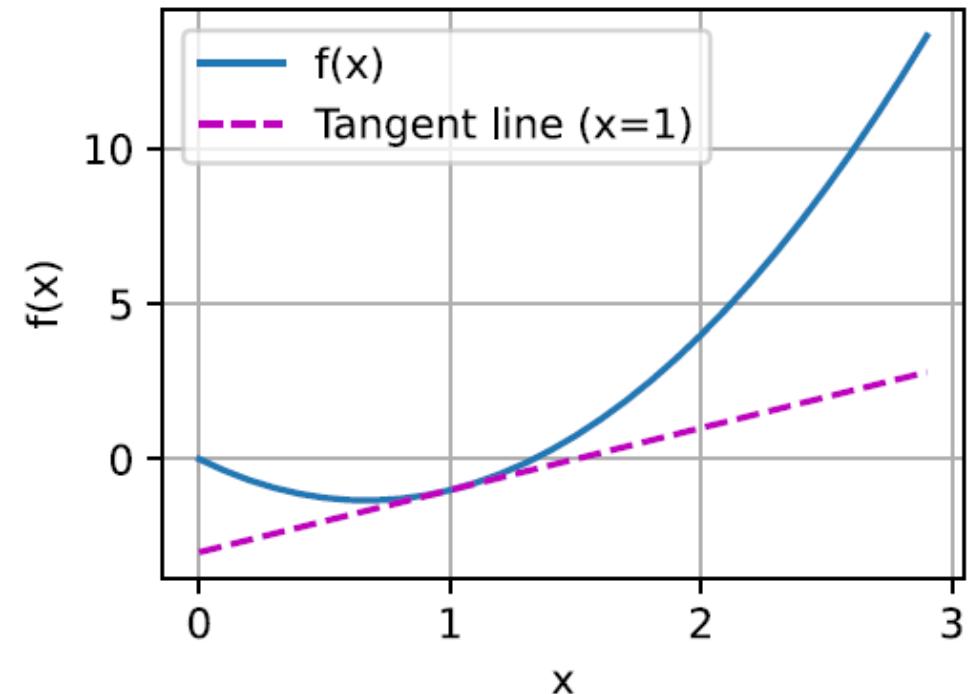
- Derivative of a function  $f(x)$

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

- Other ways to write derivatives

Given  $y = f(x)$

$$f'(x) = y' = \frac{dy}{dx} = \frac{df}{dx} = \frac{d}{dx} f(x) = Df(x) = D_x f(x)$$



# Computing Derivatives Numerically

17

- Say we want to know  $f'(10)$ 
  - Evaluate  $f$  twice:  $f(10)$  and  $f(10 + h)$
- For  $h$  in  $[0.01, 0.001, 0.0001, 0.00001]$ 
  - Compute  $\frac{f(10+h)-f(10)}{h}$

# Compute Derivatives Symbolically

18

- This is what you learned in Calculus class

$$\frac{d}{dx} C = 0 \quad \text{for any constant } C$$

$$\frac{d}{dx} x^n = nx^{n-1} \quad \text{for } n \neq 0$$

$$\frac{d}{dx} e^x = e^x$$

$$\frac{d}{dx} \ln x = x^{-1}$$

$$\frac{d}{dx} [Cf(x)] = C \frac{d}{dx} f(x) \quad \text{Constant multiple rule}$$

$$\frac{d}{dx} [f(x) + g(x)] = \frac{d}{dx} f(x) + \frac{d}{dx} g(x) \quad \text{Sum rule}$$

$$\frac{d}{dx} [f(x)g(x)] = f(x) \frac{d}{dx} g(x) + g(x) \frac{d}{dx} f(x) \quad \text{Product rule}$$

$$\frac{d}{dx} \frac{f(x)}{g(x)} = \frac{g(x) \frac{d}{dx} f(x) - f(x) \frac{d}{dx} g(x)}{g^2(x)} \quad \text{Quotient rule}$$

- Example:

$$\frac{d}{dx} [3x^2 - 4x] = 3 \frac{d}{dx} x^2 - 4 \frac{d}{dx} x = 6x - 4.$$

# Chain Rule for Composition of Functions

19

- Given  $y = f(g(x))$  decompose to underlying functions  
 $y = f(u)$  and  $u = g(x)$

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

- Or, sometime people write

$$\frac{dy}{dx} = \frac{df}{dg} \frac{dg}{dx}$$

# Chain Rule (cont)

20

- Given  $y = h(f(x), g(x))$  then

$$\frac{dy}{dx} = \frac{dh}{df} \frac{df}{dx} + \frac{dh}{dg} \frac{dg}{dx}$$

- Use chain rule to derive:
  - Sum rule: let  $h(f, g) = f + g$
  - Product rule: let  $h(f, g) = f * g$
  - Quotient rule: let  $h(f, g) = f/g$

# Partial Derivatives

21

- Given multivariate function  $y = f(x_1, x_2, \dots, x_n)$

$$\frac{\partial y}{\partial x_i} = \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_{i-1}, x_i + h, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h}$$

- Other ways to write partial derivatives

$$\frac{\partial y}{\partial x_i} = \frac{\partial f}{\partial x_i} = \partial_{x_i} f = \partial_i f = f_{x_i} = f_i = D_i f = D_{x_i} f.$$

# Gradient : direction of greatest change

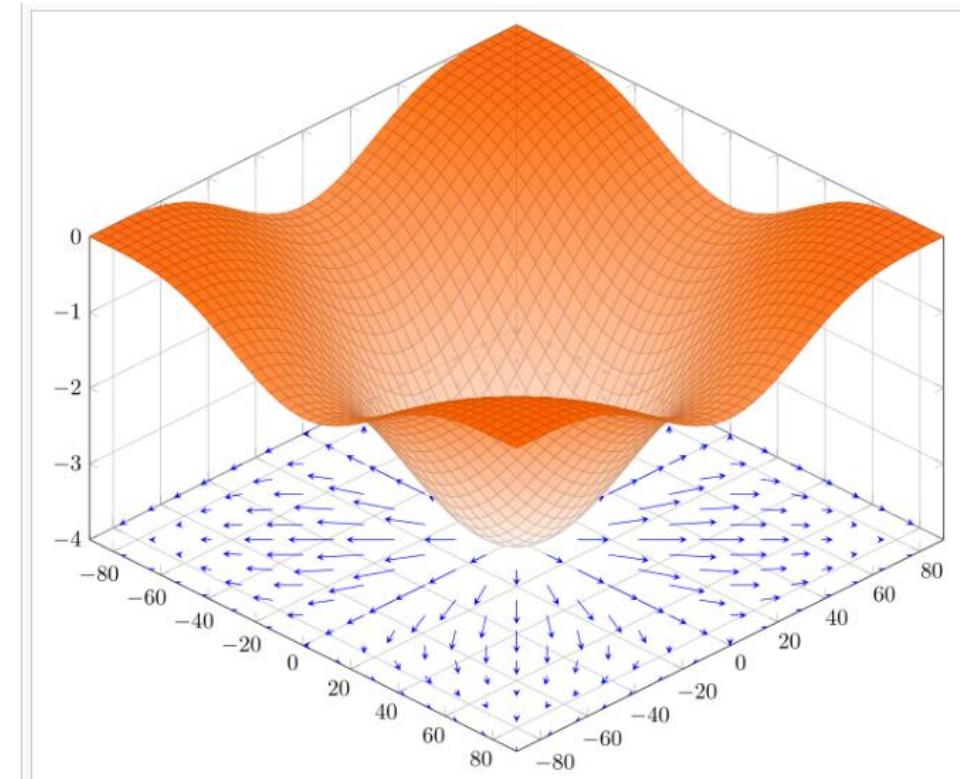
22

- Partial derivative wrt to each variable of  $f(x_1, x_2, \dots, x_n)$

- **Gradient**

$$\nabla_x f(x) = [\partial_{x_1} f(x), \partial_{x_2} f(x), \dots \partial_{x_n} f(x)]^T$$

- Gradient represent the direction of greatest change



The gradient of the function  $f(x,y) = -(\cos^2 x + \cos^2 y)^2$  depicted as a projected vector field on the bottom plane.

# Chain Rule for Multivariate Functions

23

- Suppose  $y = f(\mathbf{u})$  has variable  $u_1, u_2, \dots, u_m$  where  $u_i = g_i(\mathbf{x})$  has variables  $x_1, x_2, \dots, x_n$ , i.e.,  $\mathbf{u} = g(\mathbf{x})$

- Chain rule

$$\frac{\partial y}{\partial x_i} = \frac{\partial y}{\partial u_1} \frac{\partial u_1}{\partial x_i} + \frac{\partial y}{\partial u_2} \frac{\partial u_2}{\partial x_i} + \dots + \frac{\partial y}{\partial u_m} \frac{\partial u_m}{\partial x_i}$$

- Rewriting using vectors

$$\frac{\partial y}{\partial x_i} = \left( \frac{\partial u_1}{\partial x_i}, \dots, \frac{\partial u_m}{\partial x_i} \right) \begin{pmatrix} \frac{\partial y}{\partial u_1} \\ \vdots \\ \frac{\partial y}{\partial u_m} \end{pmatrix}$$

# Chain Rule for Multivariate Functions

24

- Gradient for multivariate functions

$$\nabla_x y = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \dots & \frac{\partial u_m}{\partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial u_1}{\partial x_n} & \dots & \frac{\partial u_m}{\partial x_n} \end{pmatrix} \begin{pmatrix} \frac{\partial y}{\partial u_1} \\ \vdots \\ \frac{\partial y}{\partial u_m} \end{pmatrix} = J^T \nabla_u y$$

- Where  $J$  is called the Jacobian Matrix

# Automatic Differentiation

25

- Problem with previous methods
  - ▣ Symbolic differentiation is too difficult if the function is a computer program
  - ▣ Numerical differentiation requires discretization which introduce numerical errors
  - ▣ Both methods are slow for multivariate functions
- Key observation: computer program is composed of elementary function
  - ▣ From elementary functions ( $\exp$ ,  $\log$ ,  $\sin$ ,  $\cos$ )
  - ▣ Using addition, multiplication, division and function composition
- But we know how to differentiate elementary functions, and we have differential rules (sum, product, quotient, chain rules)

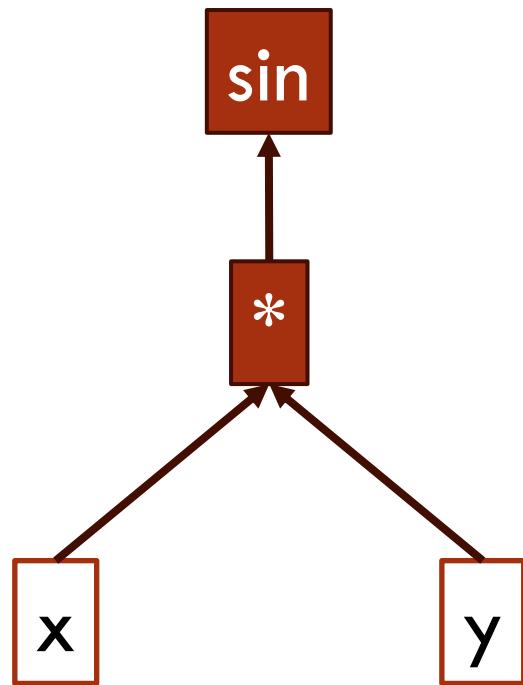
Symbolic differentiation & Numeric differentiation

chapter\_preliminaries/autograd.ipynb

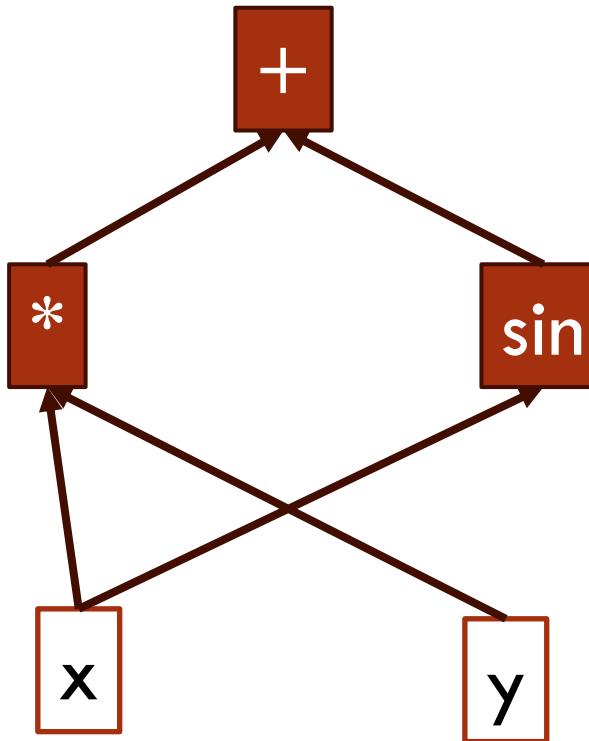
# Parse Tree

26

□  $\sin(xy)$



□  $xy + \sin(x)$



# Forward Accumulation Example: $\sin(xy)$

- Differentiate wrt  $x$  at  $x = 2, y = 3$ :

$$\frac{\partial}{\partial x} [\sin(xy)]$$

Apply chain rule and evaluate

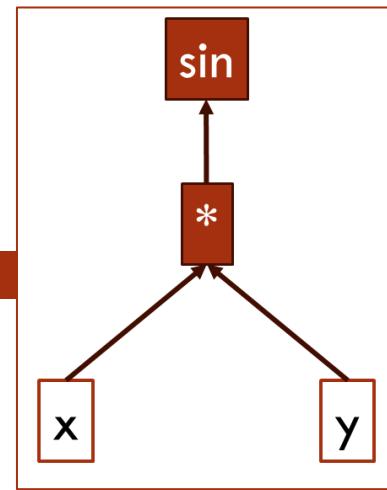
$$= \cos(xy) \frac{\partial}{\partial x} [xy] = \cos(2 * 3) \frac{\partial}{\partial x} [xy]$$

Apply product rule and evaluate

$$= \cos(6) \left[ y \frac{\partial}{\partial x} x + x \frac{\partial}{\partial x} y \right] = \cos(6) \left[ 3 \frac{\partial}{\partial x} x + 2 \frac{\partial}{\partial x} y \right]$$

$$= \cos(6) * 3 * 1$$

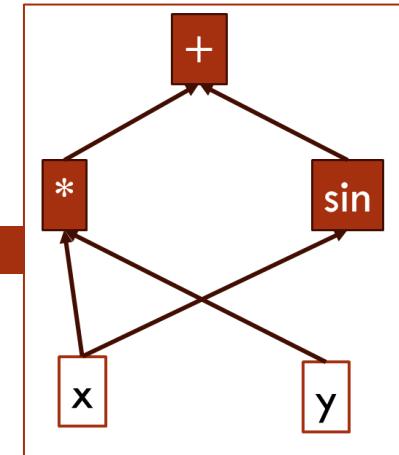
Multiple all the factors



# Forward Accumulation: $xy + \sin(x)$

28

- Differentiate wrt to  $x$  at  $x = 2, y = 3$ :



$$\frac{\partial}{\partial x} [xy + \sin(x)]$$

Apply sum rule and evaluate

$$= \frac{\partial}{\partial x} [xy] + \frac{\partial}{\partial x} [\sin(x)] = 1 \frac{\partial}{\partial x} [xy] + 1 \frac{\partial}{\partial x} [\sin(x)]$$

Apply product rule and evaluate

$$= 1 \left[ y \frac{\partial}{\partial x} x + x \frac{\partial}{\partial x} y \right] + 1 \left[ \cos(x) \frac{\partial}{\partial x} x \right] = 1 \left[ 3 \frac{\partial}{\partial x} x + 2 \frac{\partial}{\partial x} y \right] + 1 \left[ \cos(2) \frac{\partial}{\partial x} x \right]$$

Apply chain rule and evaluate

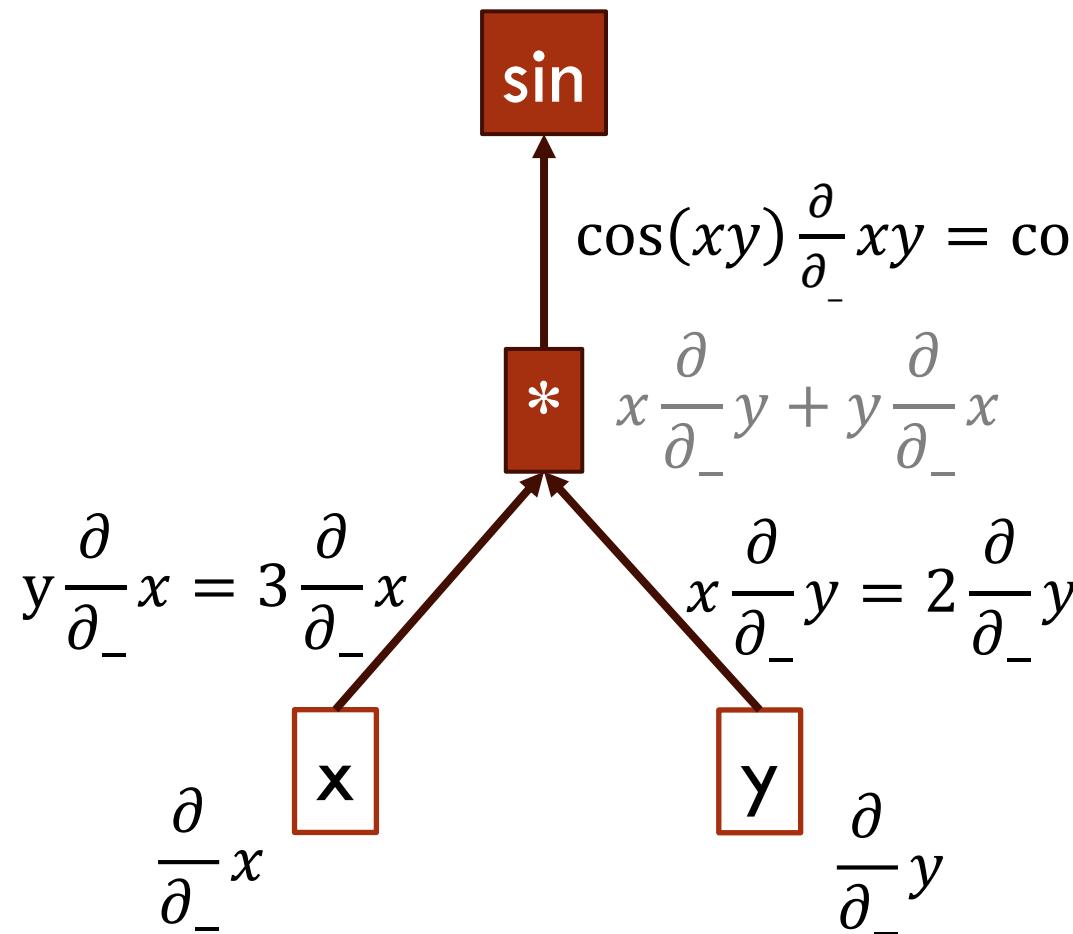
$$= 1 * 3 + 1 * \cos(2)$$

Multiple all the factors, then add

# Backward Accumulation: $\sin(xy)$

29

- At  $x = 2, y = 3$  differentiate  $\frac{\partial}{\partial_-} [\sin xy]$



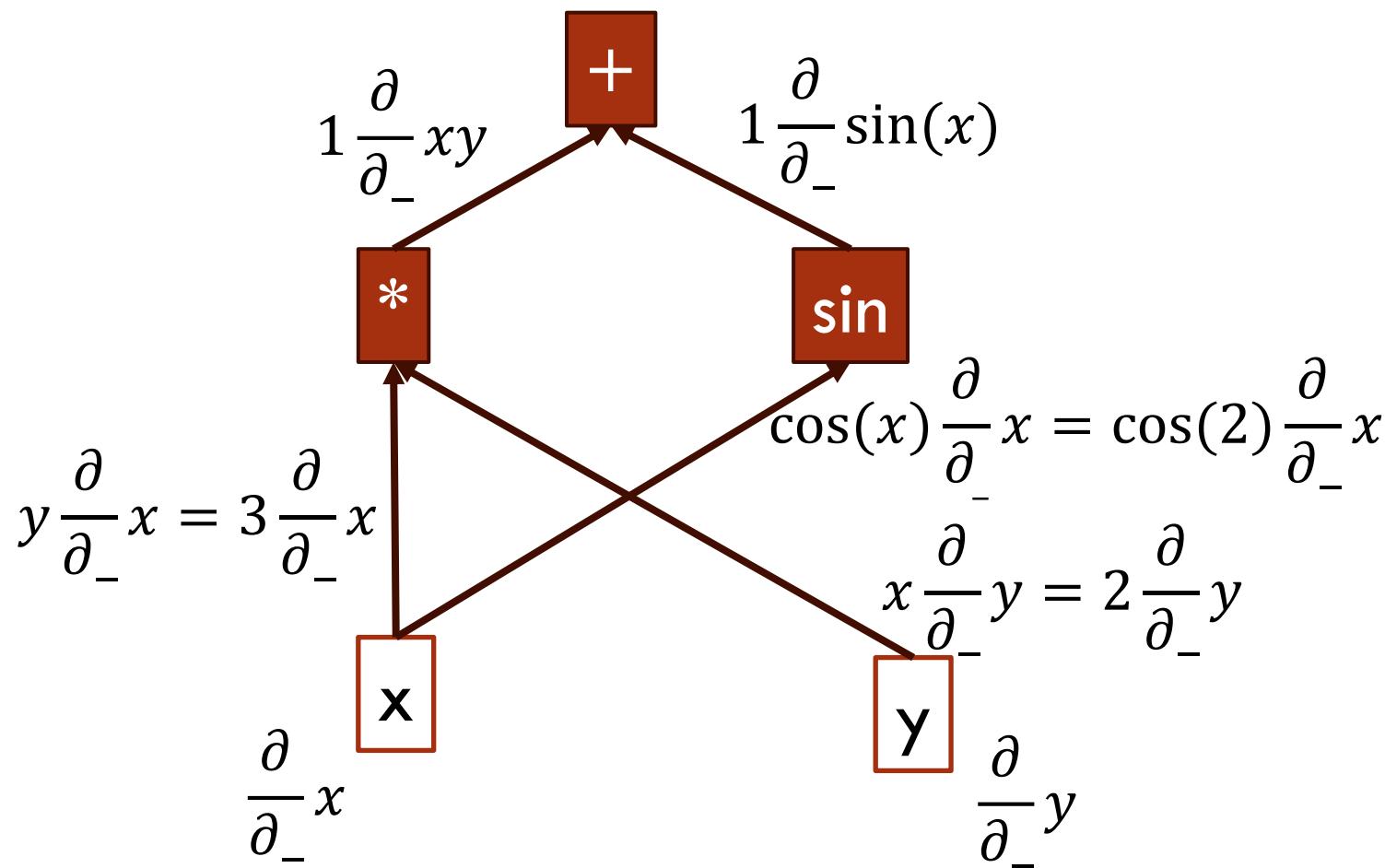
$$\frac{\partial}{\partial x} \sin(xy) = \cos(6)(3)$$

$$\frac{\partial}{\partial y} \sin(xy) = \cos(6)(2)$$

# Backward Accumulation: $xy + \sin(x)$

30

- At  $x = 2, y = 3$  differentiate  $\frac{\partial}{\partial_-} \{xy + \sin(x)\}$



$$\frac{\partial}{\partial y} xy + \sin(x) = 2$$

$$\frac{\partial}{\partial x} xy + \sin(x) = 3 + \cos(2)$$

# Automatic Differentiation Backward Accumulation with Python

31

```
class Var:

    def __init__(self, value, children=None, op='Var'):
        self.value = value
        self.grad = 0
        self.children = children or []
        self.op = op
    def __add__(self, other):
        return Var(self.value + other.value,
                   [(1, self), (1, other)], 'add')
    def __mul__(self, other):
        return Var(self.value * other.value,
                   [(other.value, self), (self.value, other)], 'mul')
    def sin(self):
        return Var(math.sin(self.value),
                   [(math.cos(self.value), self)], 'sin')
    def backward(self, grad=1):
        self.grad += grad
        for coef, child in self.children:
            child.backward(grad * coef)
    def __repr__(self):
        return f'({self.op}, {self.value})'
```

```
# Example: f(x, y) = x * y + sin(x)
x = Var(2, op='x')
y = Var(3, op='y')
z = x * y + x.sin()
```

- See Brightspace
  - [content/notebooks/AutomaticDifferentiationBackwardAccumulation.ipynb](#)
  - [content/notebooks/AutomaticDifferentiationForwardAccumulation.ipynb](#)

# Automatic Differentiation

33

- Previous examples use scalar functions:  $f: \mathcal{R}^n \rightarrow \mathcal{R}$  *loss*
- Automatic differentiation can be extended to vector functions:  
 $f: \mathcal{R}^n \rightarrow \mathcal{R}^m$  *weights*
- Forward accumulation differentiation is more efficient if  $n \ll m$
- Backward accumulation differentiation is more efficient if  $n \gg m$
- In deep learning
  - The output is usually a scalar loss function,  $m = 1$
  - The input is huge, i.e.,  $n$  is the number of weights in the neural network



SKIP

# Observable and Unobservable Variables

35

- Data usually comes from a process that is not completely known
- We model this as a random process and analyze it using probability theory
- Example
  - Tossing a coin as a random process, as we cannot predict at any toss on which side the coin will land. Heads or Tails?
  - We can only talk about the probability of the outcome (*observable variable*) for the next toss.
  - In case we have all the information (e.g. speed of throw, angle, wind, etc) we might be able to precisely predict outcome. These are the *unobservable variables*.

# Random Variable

36

- A *random variable*  $X$  is a function that assigns a number to each outcome in the sample space of a random experiment
- The probability of a particular outcome  $x$  is defined by probability distribution  $P(X=x)$
- Coin toss example:
  - Let  $X=1$  denote the outcome is Heads, and  $X=0$  is Tails
    - Random variable  $X \in \{1,0\}$
  - Let  $p_0$  be the probability the outcome is Heads:  
$$P(X = 1) = p_0 \text{ and } P(X = 0) = 1 - P(X = 1) = 1 - p_0$$
    - Bernoulli:  $P(X = i) = p_o^i(1 - p_o)^{1-i}$

# Sample

37

- If we do not know  $P(X)$ , then we need a sample  $\mathcal{X}$  (e.g. our training data) that contains examples drawn from the probability distribution of the observable variables  $x^t$  denoted as  $p(x)$ .
- The goal is to build an approximator  $\hat{p}(x)$  using sample  $\mathcal{X}$
- Coin toss example:
  - Sample:  $\mathcal{X} = \{x^t\}_{t=1}^N$   
Estimation:  $\hat{p}_0 = \# \{\text{Heads}\} / \# \{\text{Tosses}\} = \sum_t x^t / N$
  - If we observe: {heads, heads, heads, tails, heads, tails, tails, heads, heads} we have:
    - $X = \{1, 1, 1, 0, 1, 0, 0, 1, 1\}$ ,  $\hat{p}_0 = \frac{6}{9}$
  - Prediction of next toss:  
Heads if  $\hat{p}_0 > \frac{1}{2}$ , Tails otherwise

# Brief Review of Probabilities

Page 38

- Sum rule

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- Product rule

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

- If A and B are independent

$$P(A \wedge B) = P(A)P(B); \quad A \perp B$$

- Conditional independence

$$P(A \wedge B | C) = P(A|C)P(B|C); \quad A \perp B | C$$

# Bayes' Rule

Page 39

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Derivation using the product rule:

$$\begin{aligned} P(A|B) &= \frac{P(A \wedge B)}{P(B)} \\ &= \frac{P(B|A)P(A)}{P(B)} \end{aligned}$$

# Bayes' Rule

40

$$P(C | \mathbf{x}) = \frac{P(C) p(\mathbf{x} | C)}{p(\mathbf{x})}$$

posterior      prior      likelihood  
                        ↓      ↓  
                        evidence

$$P(C = 0) + P(C = 1) = 1$$

$$p(\mathbf{x}) = p(\mathbf{x} | C = 1)P(C = 1) + p(\mathbf{x} | C = 0)P(C = 0)$$

$$p(C = 0 | \mathbf{x}) + P(C = 1 | \mathbf{x}) = 1$$

# Bayes' Rule: $K > 2$ Classes

41

- In the general case we  $K$  mutually exclusive classes,  $C_i, i = 1, \dots, K$
- Prior probability satisfy:

$$P(C_i) \geq 0 \text{ and } \sum_{i=1}^K P(C_i) = 1$$

- Posterior probability of class  $C_i$  is calculated as:

$$P(C_i | \mathbf{x}) = \frac{p(\mathbf{x} | C_i)P(C_i)}{p(\mathbf{x})} = \frac{p(\mathbf{x} | C_i)P(C_i)}{\sum_{k=1}^K p(\mathbf{x} | C_k)P(C_k)}$$

- Bayes' classifier

choose  $C_i$  if  $P(C_i | \mathbf{x}) = \max_k P(C_k | \mathbf{x})$

# Example of Bayes Theorem

- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.5 \times \frac{1}{50000}}{\frac{1}{20}} = \frac{1}{5000}$$

# Another Example

Page 43

- Want to decide between two hypothesis, for a particular location
  - There is oil underground
  - There is no oil underground
- Suppose there is oil test: pos or neg

$$P(oil) = .008$$

$$P(\neg oil) = .992$$

$$P(pos | oil) = .98 \quad P(neg | oil) = .02$$

$$P(pos | \neg oil) = .03 \quad P(neg | \neg oil) = .97$$

# Another Example (cont)

- Given test is positive, compute posterior probability for oil and not oil

$$P(\text{oil} \mid \text{pos}) \propto P(\text{pos} \mid \text{oil})P(\text{oil}) = .98(.008) = .0078$$

$$P(\neg\text{oil} \mid \text{pos}) \propto P(\text{pos} \mid \neg\text{oil})P(\neg\text{oil}) = .03(.992) = .0298$$

$$P(\neg\text{oil} \mid \text{pos}) = \frac{.0298}{.0078 + .0298} = .79$$

- Given positive test, probability of not oil is 0.79