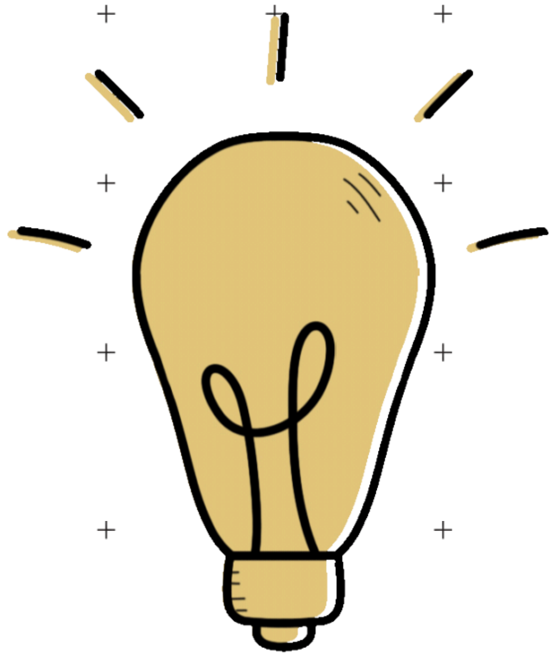
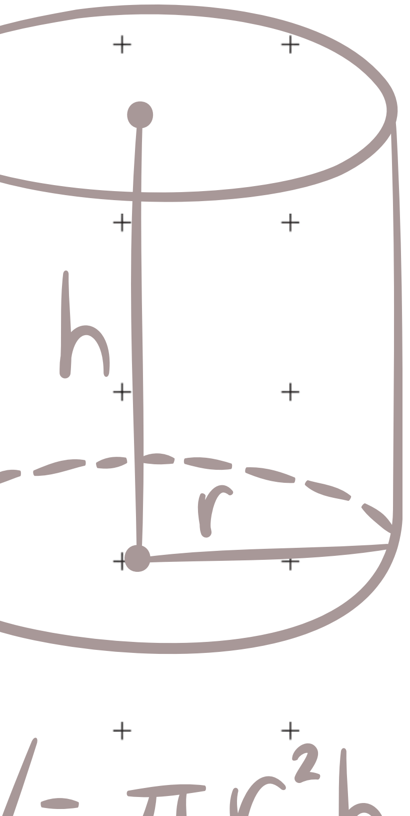


MATHEMATICAL FOUNDATION FOR COMPUTER SCIENCE

CALCULUS

Start Now



GROUP: MTNT

Nguyễn Trường Thịnh - 2213298

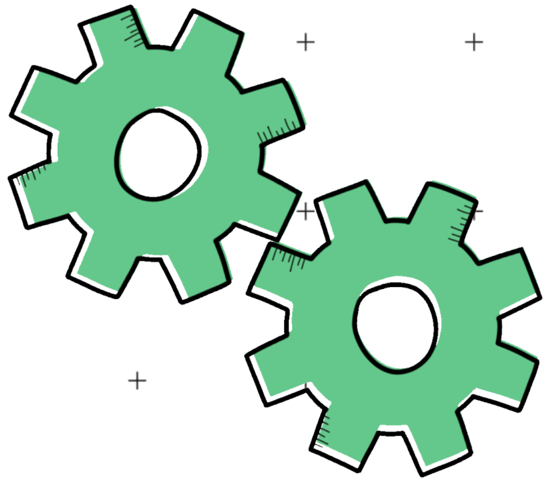
Nguyễn Hữu Nhân - 2212362

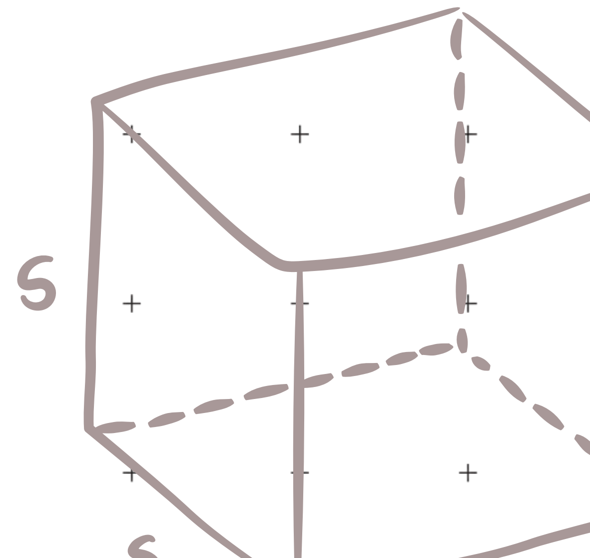
Trần Thành Tài - 2213001

Dương Quang Minh - 2212021

2.4.1

DERIVATIVES & DIFFERENTIATION





PROBLEM STATEMENT

Science/Engineering Scope:

- **Functions**: describe how one quantity changes with another.
- **Derivative of a function**: measures the <u>instantaneous rate of change</u>

Derivatives in others contexts:

- Chemical kinetics
- Computer networks
- Geometry and graphics

PROBLEM STATEMENT

How fast is this changing?

Derivative - the rate of change in a function: how a loss function increase or decrease

Derivative of f(x) at a point x:

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \quad (2.4.1)$$

PROBLEM STATEMENT

- **Problem**: Some important functions are not differentiable.
- **Challenge**: Most deep learning training algorithms require derivatives.
- **Solution**: Use a differentiable surrogate to approximate the non-differentiable function during optimization.

DERIVATIVES & DIFFERENTIATION

Derivatives notational convention:

Given (y = f(x)):

$$f'(x) = y' = \frac{dy}{dx} = \frac{df}{dx} = \frac{d}{dx}f(x) = Df(x) = D_x f(x) \quad (2.4.2)$$

The symbols $\frac{d}{dx}$ and D are differentiation operators.

DERIVATIVES & DIFFERENTIATION

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

The symbols $\frac{d}{dx}$ and D are differentiation operators.

Some common functions:

1.Constant Rule: $\frac{d}{dx}C = 0 \quad \text{for any constant } C$ (2.4.3)

2.Power Rule: $\frac{d}{dx}x^n = nx^{n-1} \quad \text{for } n \neq 0$ (2.4.4)

DERIVATIVES & DIFFERENTIATION

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

The symbols $\frac{d}{dx}$ and D are differentiation operators.

Some common functions:

3.Exponential Rule: $$\frac{d}{dx}e^x = e^x \qquad (2.4.5)$$

4.Logarithm Rule: $$\frac{d}{dx}\ln x = x^{-1} \qquad (2.4.6)$$

DERIVATIVES & DIFFERENTIATION

Compositions of differentiable functions are often differentiable.

Useful rules for composing differentiable functions and constants:

1. Constant Multiple Rule:

$$\frac{d}{dx}[Cf(x)] = C\frac{d}{dx}f(x) \tag{2.4.7}$$

2. Sum Rule: $$\frac{d}{dx}[f(x)+g(x)] = \frac{d}{dx}f(x) + \frac{d}{dx}g(x) \tag{2.4.8}$$

DERIVATIVES & DIFFERENTIATION

3.Product Rule:

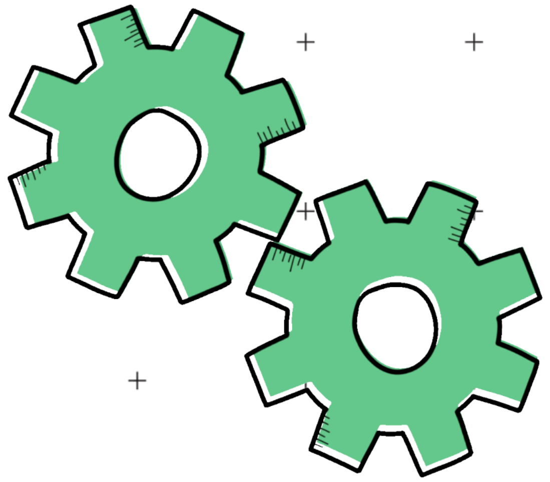
$$\frac{d}{dx}[f(x)g(x)] = f(x)\frac{d}{dx}g(x) + g(x)\frac{d}{dx}f(x) \quad (2.4.9)$$

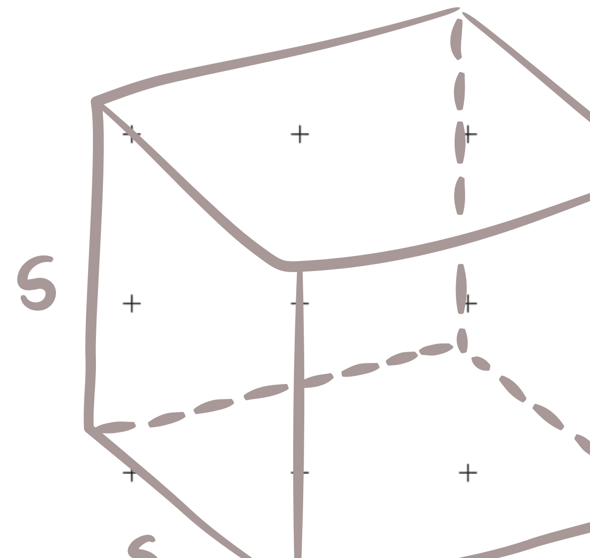
4.Quotient Rule:

$$\frac{d}{dx}\left(\frac{f(x)}{g(x)}\right) = \frac{g(x)\frac{d}{dx}f(x) - f(x)\frac{d}{dx}g(x)}{g^2(x)} \quad (2.4.10)$$

2.4.2

VISUALIZATION UTILITIES





VISUALIZATION UTILITIES

use_svg_display(): Use the svg format to display a plot in Jupyter.

set_figsize(): Set the figure size for matplotlib.

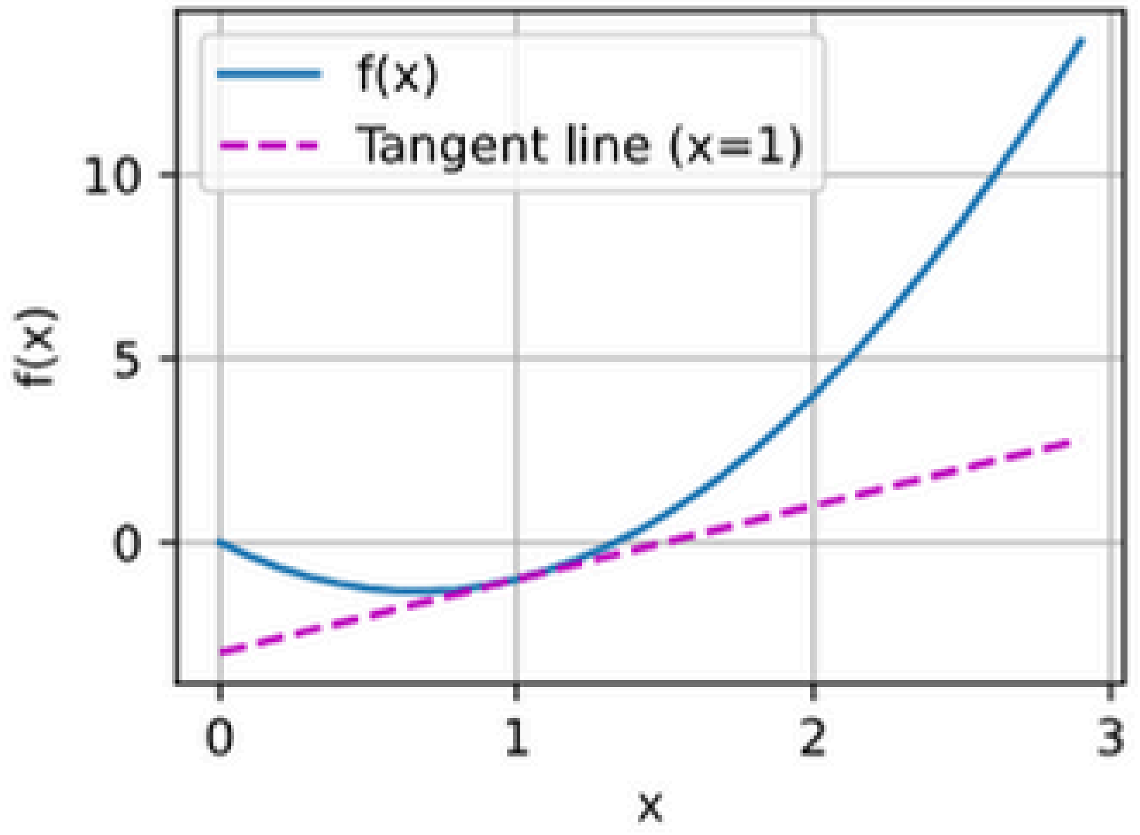
set_axes(): Set the axes for matplotlib.

plot(): Plot the function

VISUALIZATION UTILITIES

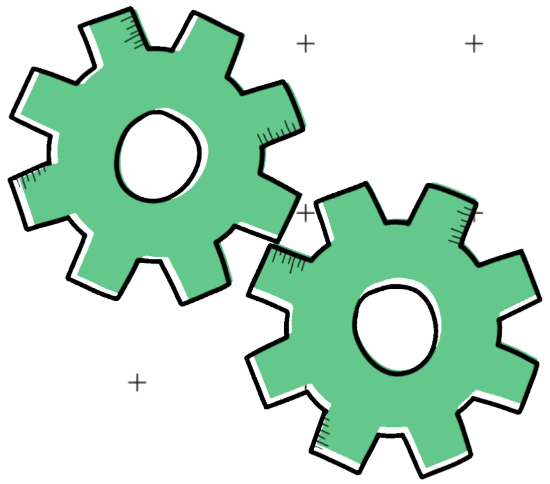
Result:

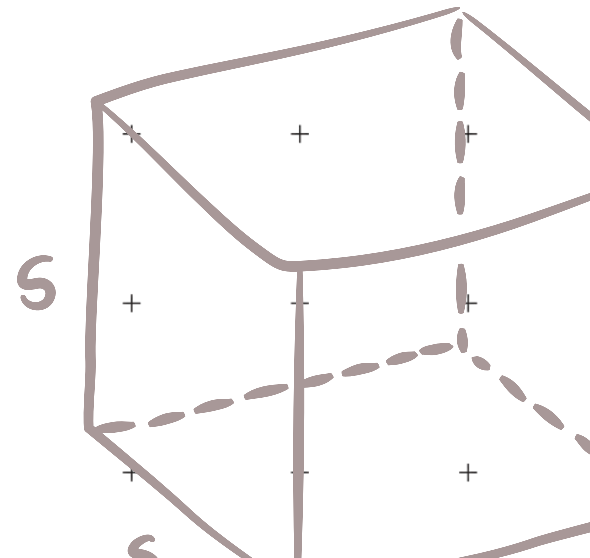
```
x = np.arange(0, 3, 0.1)
plot(x, [f(x), 2 * x - 3], 'x', 'f(x)', legend=['f(x)', 'Tangent line (x=1)'])
```



2.4.3

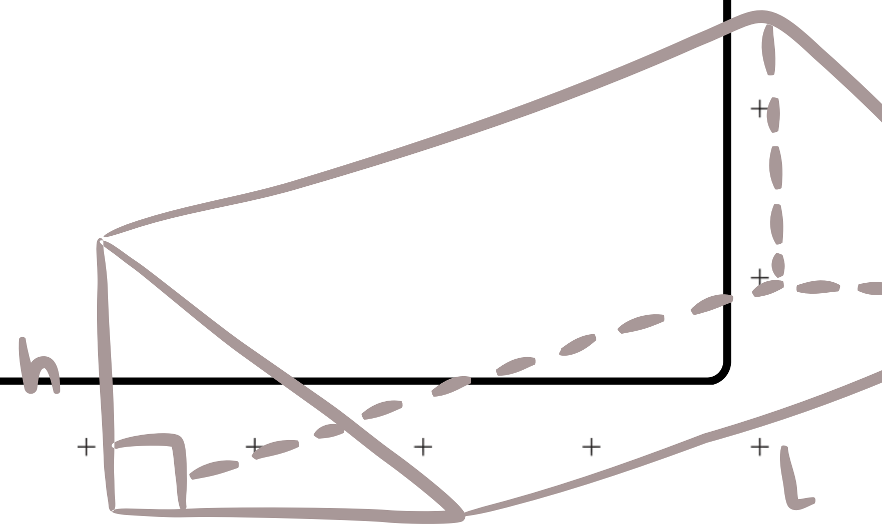
PARTIAL DERIVATIVES AND GRADIENTS





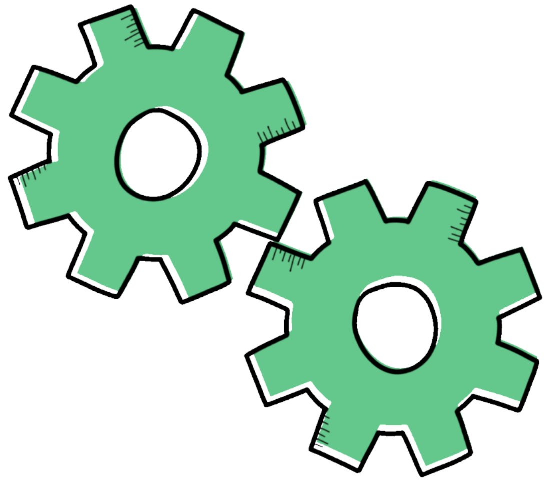
PROBLEM STATEMENT

- Many deep learning problems involve functions of multiple variables.
- Partial derivatives measure how a function changes with respect to each input variable.
- The gradient vector points in the direction of the steepest ascent.
- These tools are essential for optimization and learning algorithms.



APPLICATIONS

- Training neural networks
- Computer vision
- Natural language processing
- Robotics and control



PARTIAL DERIVATIVES

Let $y = f(x_1, x_2, \ldots, x_n)$ be a function with n variables. The partial derivative of y with respect to its i^{th} parameter x_i is:

$$\frac{\partial y}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_i, \ldots, x_n)}{h}. \quad (2.4.6)$$

The following notational conventions for partial derivatives are all common and all mean the same thing:

$$\frac{\partial y}{\partial x_i} = \frac{\partial f}{\partial x_i} = \partial_{x_i} f = \partial_i f = f_{x_i} = f_i = D_i f = D_{x_i} f. \quad (2.4.7)$$

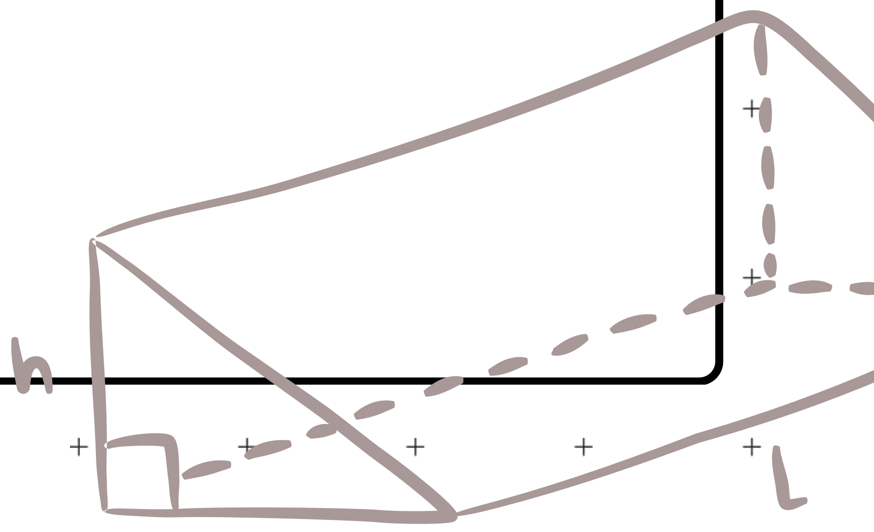
GRADIENTS

The gradient is a vector of all partial derivatives of a multivariable function.

For a function $f : R^{n} \to R$, input is an n-dimensional vector $\mathbf{x}=[x_1, \ldots, x_n]^{\top}$ output is a scalar.

The gradient of f with respect to $\mathbf{x}$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \partial_{x_1} f(\mathbf{x}) \\ \partial_{x_2} f(\mathbf{x}) \\ \vdots \\ \partial_{x_n} f(\mathbf{x}) \end{bmatrix}. \tag{2.4.8}$$



EXAMPLE

$$f(x,y) = 3x^2y + 2y^2$$

Step 1: Compute partial derivatives

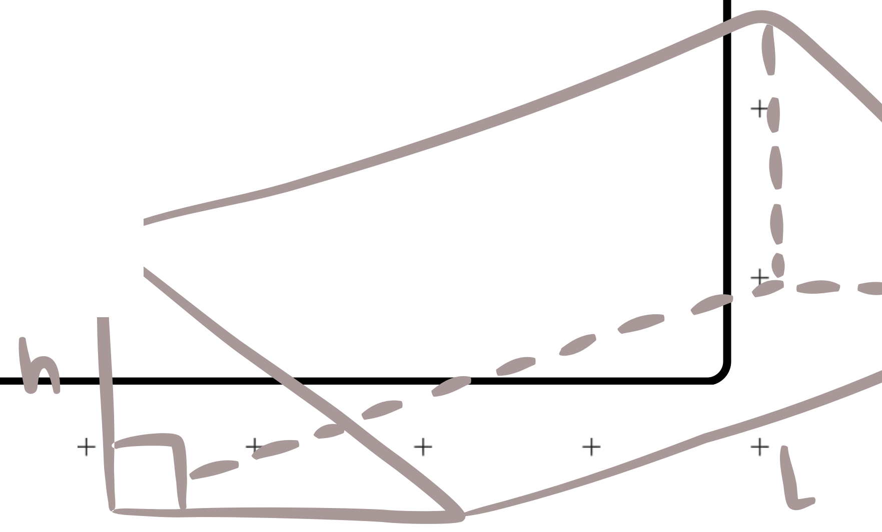
- $\frac{\partial f}{\partial x} = 6xy$
- $\frac{\partial f}{\partial y} = 3x^2 + 4y$

Step 2: Form the gradient vector

$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 6xy \\ 3x^2 + 4y \end{bmatrix}$$

Step 3: Evaluate the gradient at a point, x = 1, y = 2

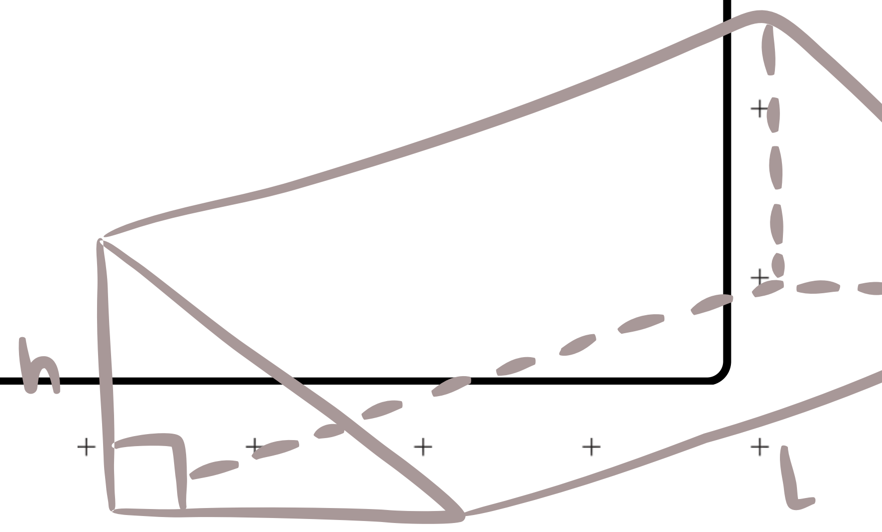
$$\nabla f(1,2) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 6.1.2 \\ 3.1^2 + 4.2 \end{bmatrix} = \begin{bmatrix} 12 \\ 11 \end{bmatrix}$$



EXAMPLE

```python
import sympy as sp

# Define symbols
x, y = sp.symbols('x y')

# Define the function
f = 3 * x**2 * y + 2 * y**2

# Compute partial derivatives
df_dx = sp.diff(f, x)
df_dy = sp.diff(f, y)

# Display gradients
print("∂f/∂x:", df_dx)
print("∂f/∂y:", df_dy)

gradient_at_point = [df_dx.subs({x: 1, y: 2}), df_dy.subs({x: 1, y: 2})]
print("Gradient at (1,2):", gradient_at_point)
```

```
∂f/∂x: 6xy
∂f/∂y: 3x^2 + 4y
Gradient at (1,2): [12, 11]
```



RULES FOR DIFFERENTIATING MULTIVARIABLE FUNCTIONS

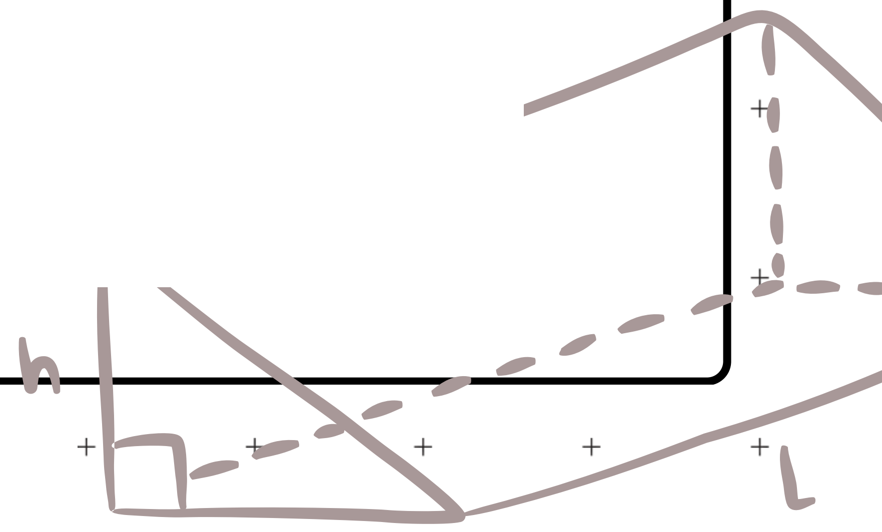
Rule 1: For all $\mathbf{A} \in \mathbb{R}^{m \times n}$ we have $\nabla_{\mathbf{x}}(\mathbf{A}\mathbf{x}) = \mathbf{A}^T$ and $\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{A}) = \mathbf{A}$.

We have:

$$\mathbf{A} = \begin{bmatrix} A_{00} & A_{01} & \dots & A_{0n} \\ A_{10} & A_{11} & \dots & A_{1n} \\ & & \vdots & \\ A_{m0} & A_{m1} & \dots & A_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

Then

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} A_{00}x_0 + A_{01}x_1 + \cdots + A_{0n}x_n \\ A_{10}x_0 + A_{11}x_1 + \cdots + A_{1n}x_n \\ \vdots \\ A_{n0}x_0 + A_{n1}x_1 + \cdots + A_{nn}x_n \end{bmatrix}$$



RULES FOR DIFFERENTIATING MULTIVARIABLE FUNCTIONS

Rule 1: For all $\mathbf{A} \in \mathbb{R}^{m\times n}$ we have $\nabla_{\mathbf{x}}(\mathbf{A}\mathbf{x}) = \mathbf{A}^T$ and $\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{A}) = \mathbf{A}$.

Set $f = \mathbf{A}\mathbf{x}$, then

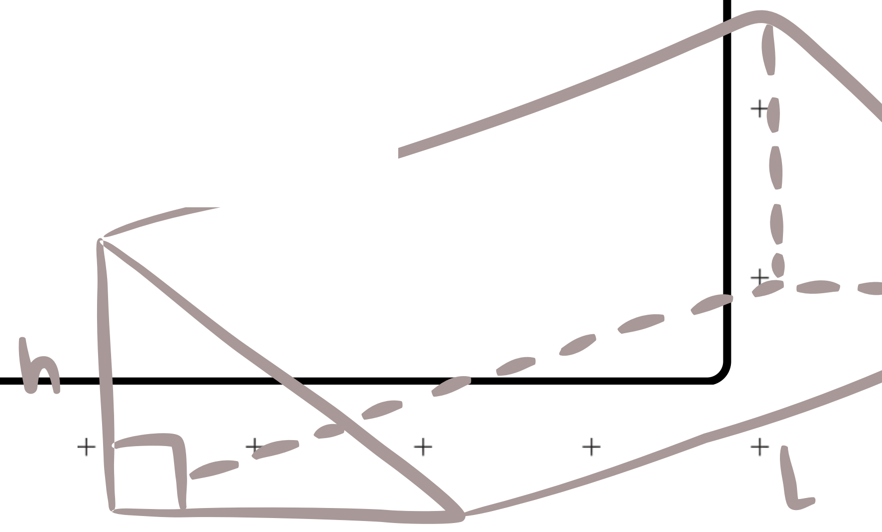
$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \partial_{x_0} f_0(\mathbf{x}) & \partial_{x_1} f_0(\mathbf{x}) & \ldots & \partial_{x_n} f_0(\mathbf{x}) \\ \partial_{x_0} f_1(\mathbf{x}) & \partial_{x_1} f_1(\mathbf{x}) & \ldots & \partial_{x_n} f_1(\mathbf{x}) \\ & & \vdots & \\ \partial_{x_0} f_n(\mathbf{x}) & \partial_{x_1} f_n(\mathbf{x}) & \ldots & \partial_{x_n} f_n(\mathbf{x}) \end{bmatrix}^T = \begin{bmatrix} A_{00} & A_{01} & \ldots & A_{0n} \\ A_{10} & A_{11} & \ldots & A_{1n} \\ & & \vdots & \\ A_{m0} & A_{m1} & \ldots & A_{mn} \end{bmatrix}^T = \mathbf{A}^T$$

So for all $\mathbf{A} \in \mathbb{R}^{m\times n}$, we have

$$\nabla_{\mathbf{x}}(\mathbf{A}\mathbf{x}) = \mathbf{A}^T.$$

Similarly for all $\mathbf{A} \in \mathbb{R}^{n\times m}$, we have

$$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{A}) = \mathbf{A}.$$



RULES FOR DIFFERENTIATING MULTIVARIABLE FUNCTIONS

Rule 2: For square matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$: $\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{A}\mathbf{x}) = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}$, in particular $\nabla_{\mathbf{x}}\|\mathbf{x}\|^2 = 2\mathbf{x}$.

Let $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n x_i A_{ij} x_j$, then

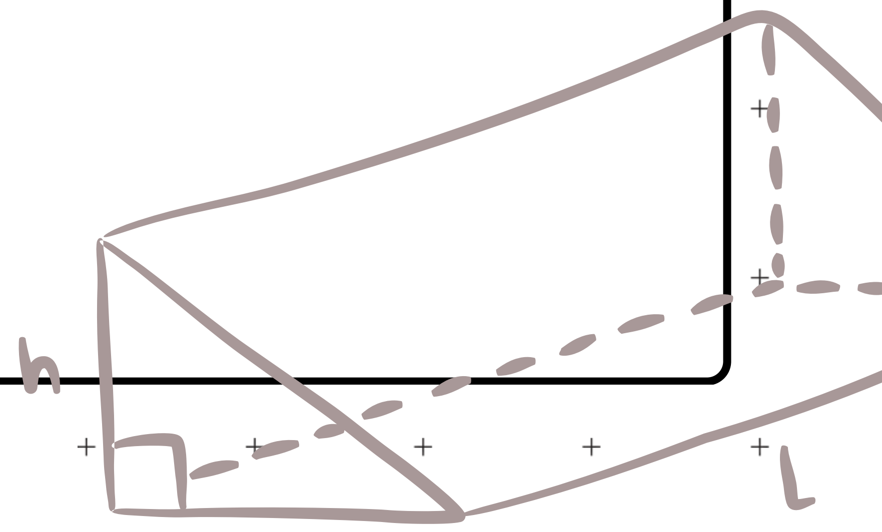
$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \partial_{x_1} f(\mathbf{x}) \\ \partial_{x_2} f(\mathbf{x}) \\ \vdots \\ \partial_{x_n} f(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n \frac{\partial x_1 A_{1j} x_j}{\partial x_1} + \sum_{i=1}^n \frac{\partial x_i A_{i1} x_1}{\partial x_1} \\ \sum_{j=1}^n \frac{\partial x_2 A_{2j} x_j}{\partial x_2} + \sum_{i=1}^n \frac{\partial x_i A_{i2} x_2}{\partial x_2} \\ \vdots \\ \sum_{j=1}^n \frac{\partial x_n A_{nj} x_j}{\partial x_n} + \sum_{i=1}^n \frac{\partial x_i A_{in} x_n}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n A_{1j} x_j + \sum_{i=1}^n x_i A_{i1} \\ \sum_{j=1}^n A_{2j} x_j + \sum_{i=1}^n x_i A_{i2} \\ \vdots \\ \sum_{j=1}^n A_{nj} x_j + \sum_{i=1}^n x_i A_{in} \end{bmatrix}$$

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \sum_{j=1}^n A_{1j} x_j \\ \sum_{j=1}^n A_{2j} x_j \\ \vdots \\ \sum_{j=1}^n A_{nj} x_j \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^n x_i A_{i1} \\ \sum_{i=1}^n x_i A_{i2} \\ \vdots \\ \sum_{i=1}^n x_i A_{in} \end{bmatrix} = \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}.$$

Similarly, for any matrix $\mathbf{X}$, we have $\|\mathbf{X}\|_F^2 = \mathbf{X}^T\mathbf{X} = \mathbf{X}^T\mathbf{I}\mathbf{X}$, with $\mathbf{I}$ is Identity matrix.

Then, apply Rule 2, we have $\|\mathbf{X}\|_F^2 = \mathbf{X}^T\mathbf{I}\mathbf{X} = (\mathbf{I} + \mathbf{I}^T)\mathbf{X} = 2\mathbf{I}\mathbf{X} = 2\mathbf{X}$

So, $\nabla_X \|X\|_F^2 = 2X$.



2.4.4

CHAIN RULE

INTRODUCTION

The Chain Rule is a mathematical tool used to compute the derivative of composite functions.

There are two main cases:

- **Single-variable functions:** $y=f(g(x))$
- **Multivariable functions:** $y=f(u)$, where $u=g(x)$

CHAIN RULE FOR SINGLE-VARIABLE FUNCTIONS

Suppose that $y=f(f(x))$ and that the underlying functions $y=f(u)$ and $u=g(x)$ are both differentiable. The chain rule states that

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx} \qquad (2.4.9)$$

CHAIN RULE FOR SINGLE-VARIABLE FUNCTIONS

Example: Find the derivative of the function $y=\sin(x^2)$.

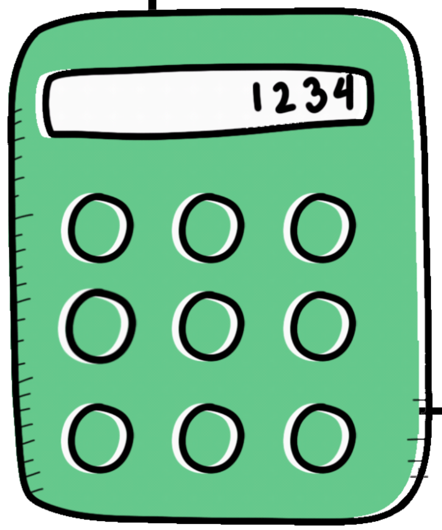
Let $u = x^2$, then $y = \sin(u)$.

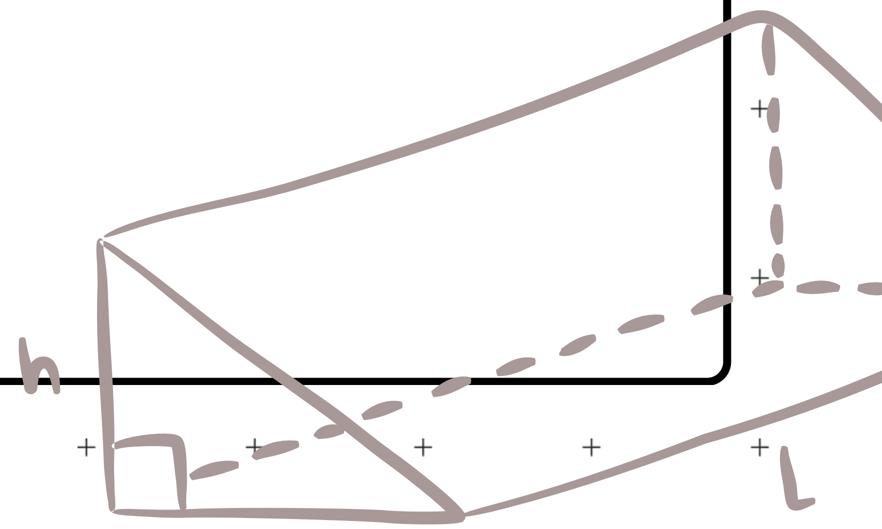
Differentiate each part:

- $\frac{dy}{du} = \cos(u)$
- $\frac{du}{dx} = 2x$

By applying the Chain Rule:

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx} = \cos(u) \cdot 2x = \cos(x^2) \cdot 2x$$

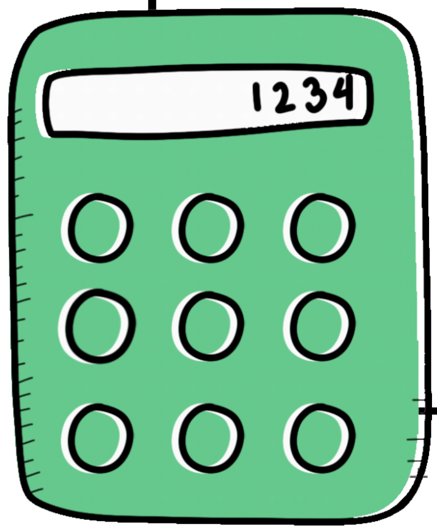


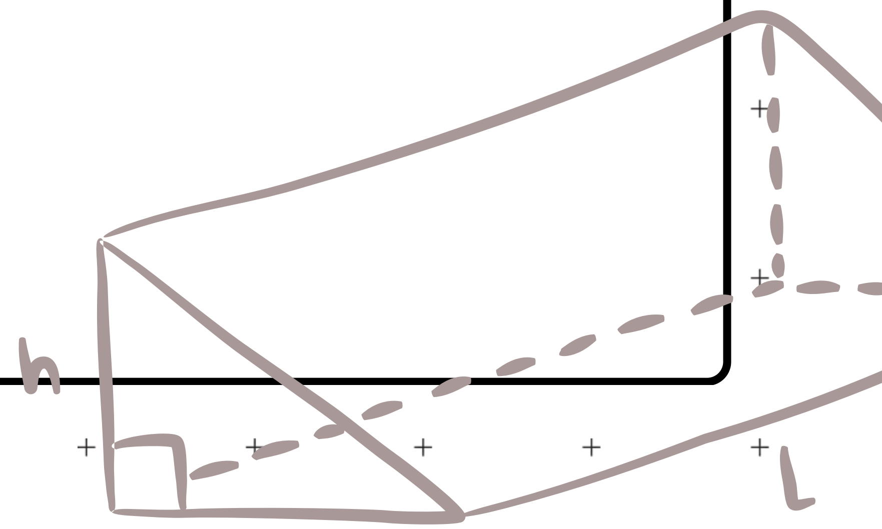


CHAIN RULE FOR MULTIVARIABLE FUNCTIONS

Suppose that $y=f(\mathbf{u})$ has variables $u_1, u_2, \ldots u_m$ and each $u_i = g_i(\mathbf{x})$ has variables $x_1, x_2, \ldots x_n$. $\mathbf{u}=g(\mathbf{x})$. The chain rule states that

$$\frac{\partial y}{\partial x_i} = \frac{\partial y}{\partial u_1}\frac{\partial u_1}{\partial x_i} + \frac{\partial y}{\partial u_2}\frac{\partial u_2}{\partial x_i} + \cdots + \frac{\partial y}{\partial u_m}\frac{\partial u_m}{\partial x_i} \qquad (2.4.10)$$





CHAIN RULE FOR MULTIVARIABLE FUNCTIONS

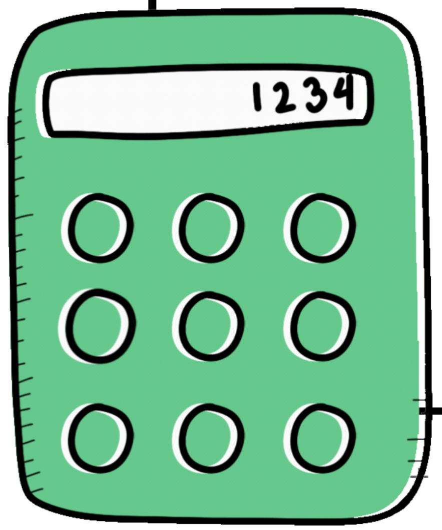
Because that:

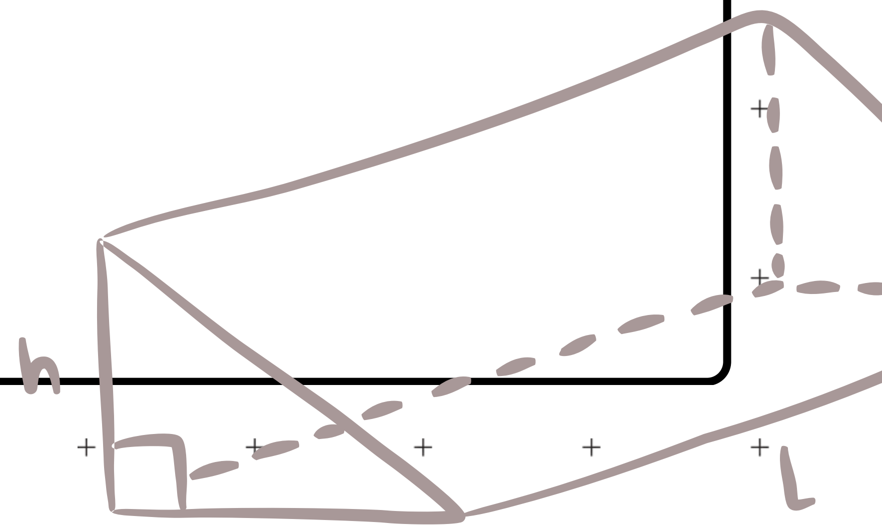
$$\nabla_{\mathbf{x}} y = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} \in \mathbb{R}^n \qquad \nabla_{\mathbf{u}} y = \begin{bmatrix} \frac{\partial y}{\partial u_1} \\ \frac{\partial y}{\partial u_2} \\ \vdots \\ \frac{\partial y}{\partial u_m} \end{bmatrix} \in \mathbb{R}^m$$

Then

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

$$\begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_2}{\partial x_1} & \cdots & \frac{\partial u_m}{\partial x_1} \\ \frac{\partial u_1}{\partial x_2} & \frac{\partial u_2}{\partial x_2} & \cdots & \frac{\partial u_m}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial u_1}{\partial x_n} & \frac{\partial u_2}{\partial x_n} & \cdots & \frac{\partial u_m}{\partial x_n} \end{bmatrix} * \begin{bmatrix} \frac{\partial y}{\partial u_1} \\ \frac{\partial y}{\partial u_2} \\ \vdots \\ \frac{\partial y}{\partial u_m} \end{bmatrix}$$





CHAIN RULE FOR MULTIVARIABLE FUNCTIONS

So Final Vectorized Chain Rule:

$$\nabla_{\mathbf{x}} y = \mathbf{A} \nabla_{\mathbf{u}} y$$

Where:

- $\mathbf{A} \in \mathbb{R}^{n \times m}$ is a matrix containing the partial derivatives $\frac{\partial u_j}{\partial x_i}$.
- $\nabla_{\mathbf{x}} y$: Gradient of y with respect to $\mathbf{x}$.
- $\nabla_{\mathbf{u}} y$: Gradient of y with respect to $\mathbf{u}$.

CHAIN RULE FOR MULTIVARIABLE FUNCTIONS

Example: Suppose $y=u_1^2+u_2^2$; $u_1=x_1+x_2$; $u_2=x_1-x_2$. Compute $\frac{\partial y}{\partial x_1}$

Compute the partial derivatives

$$\frac{\partial y}{\partial u_1} = 2u_1, \quad \frac{\partial y}{\partial u_2} = 2u_2, \quad \frac{\partial u_1}{\partial x_1} = 1, \quad \frac{\partial u_2}{\partial x_1} = 1$$

Apply chain rule and substitute

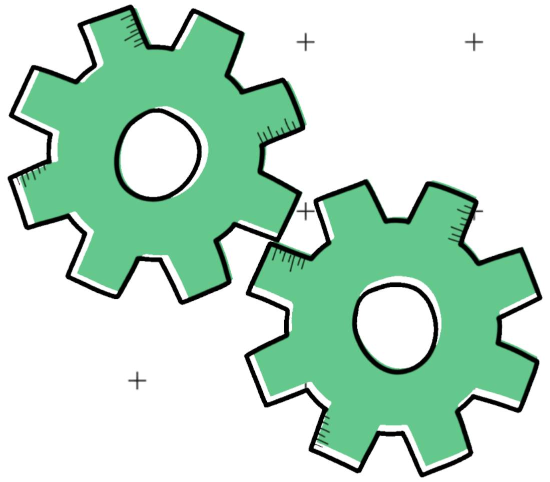
$$\frac{\partial y}{\partial x_1} = \frac{\partial y}{\partial u_1} \cdot \frac{\partial u_1}{\partial x_1} + \frac{\partial y}{\partial u_2} \cdot \frac{\partial u_2}{\partial x_1} = (2u_1) \cdot 1 + (2u_2) \cdot 1 = 2u_1 + 2u_2$$

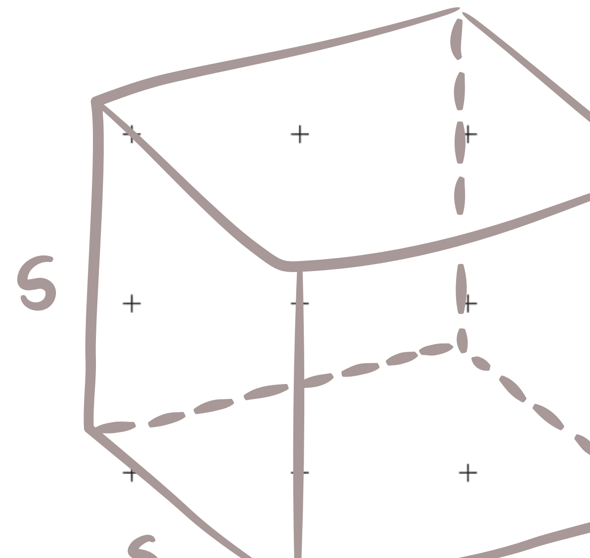
Substitute $u_1 = x_1 + x_2, u_2 = x_1 - x_2$:

$$\frac{\partial y}{\partial x_1} = 2(x_1 + x_2) + 2(x_1 - x_2) = 4x_1$$

2.4.5

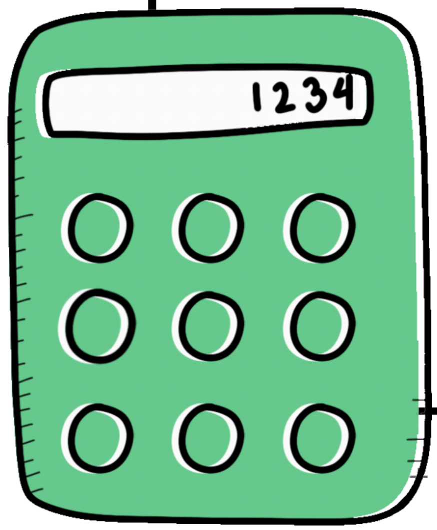
APPLICATION

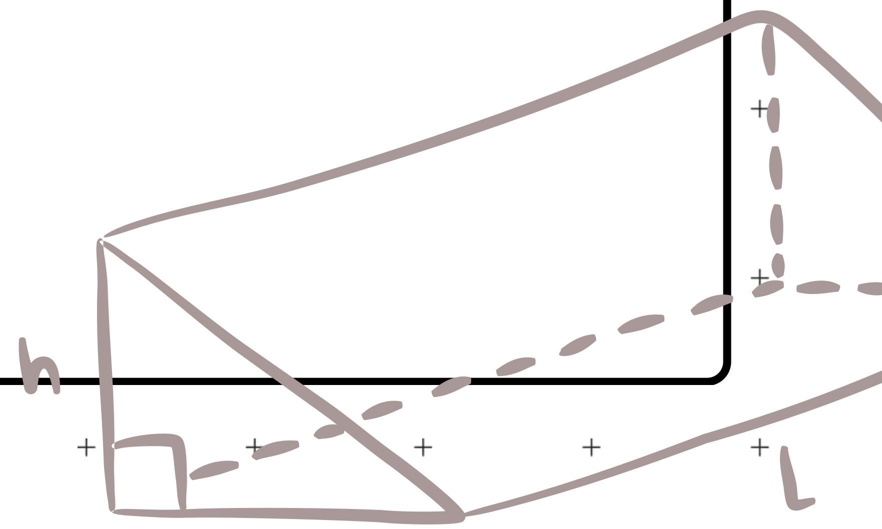




CONVOLUTIONAL NEURAL NETWORK

- In a Convolutional Neural Network (CNN), filters (kernels) like matrix W are trainable parameters.
- Goal: Adjust W so that the network output matches a target y as closely as possible.
- Question: How does a CNN learn the right values of W?





PROBLEM SETUP

We use a simple 2×2 input and 2×2 kernel to demonstrate learning in CNN.

Input:

$$I = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

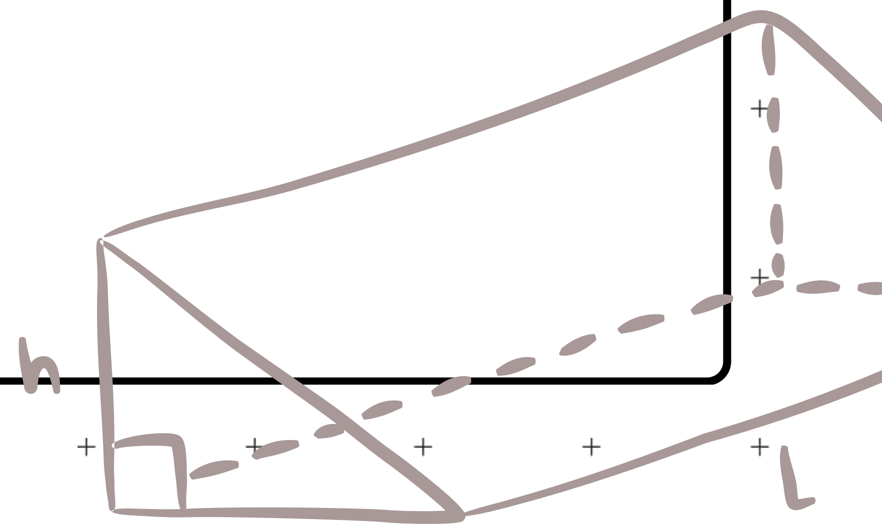
Kernel (weights)**:**

$$W = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$$

Target output: y = 30

Loss function:

$$L = (O - y)^2$$



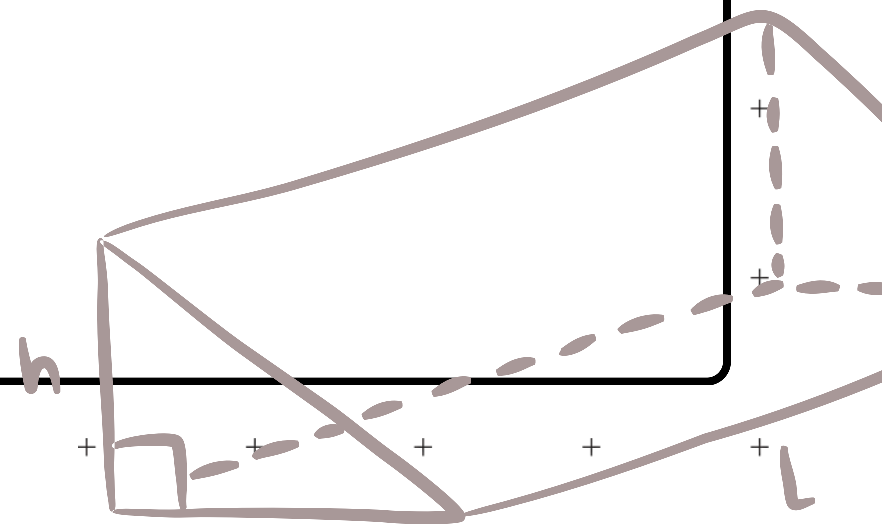
FORWARD PROPAGATION

Perform convolution (no padding, stride 1 → output is a scalar):

$$O = \sum I_{ij} \cdot W_{ij} = 1 \cdot 0 + 2 \cdot 1 + 3 \cdot 2 + 4 \cdot 3 = 20$$

Compute loss:

$$L = (20 - 30)^2 = 100$$



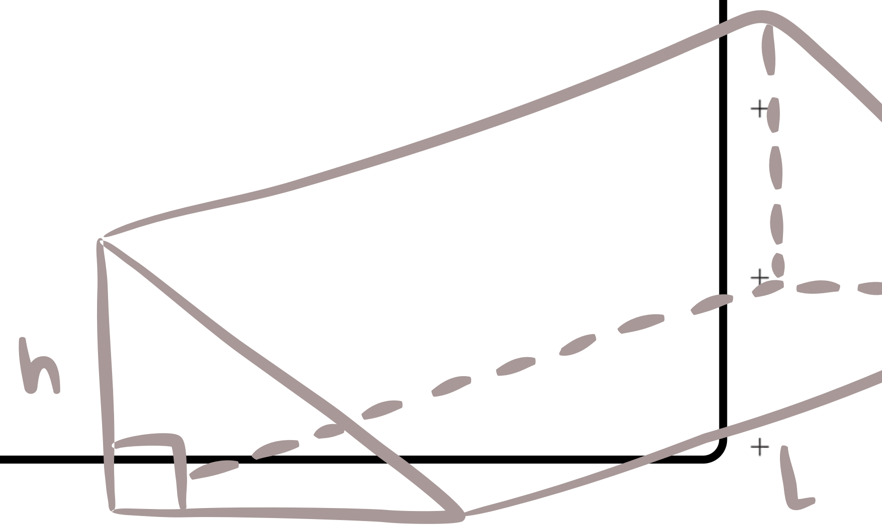
BACKPROPAGATION — COMPUTE GRADIENTS

Gradient of loss with respect to the output:

$$\frac{\partial L}{\partial O} = 2(O - y) = 2(20 - 30) = -20$$

Gradient of the output with respect to each weight W:

$$O = \sum_{i,j} I_{ij} \cdot W_{ij} \Rightarrow \frac{\partial O}{\partial W_{ij}} = I_{ij}$$



GRADIENT DESCENT UPDATE

Therefore, gradient of the loss with respect to the kernel:

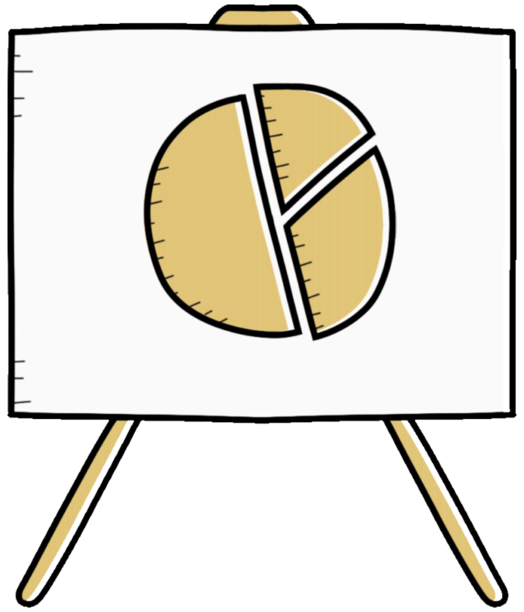
$$\frac{\partial L}{\partial W_{ij}} = \frac{\partial L}{\partial O} \cdot \frac{\partial O}{\partial W_{ij}} = -20 \cdot I_{ij}$$

$$\nabla_W L = -20 \cdot I = \begin{bmatrix} -20 & -40 \\ -60 & -80 \end{bmatrix}$$

Let learning rate η=0.01. Update rule:

$$W_{\text{new}} = W - \eta \cdot \nabla_W L$$

$$W_{\text{new}} = \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} - 0.01 \cdot \begin{bmatrix} -20 & -40 \\ -60 & -80 \end{bmatrix} = \begin{bmatrix} 0.2 & 1.4 \\ 2.6 & 3.8 \end{bmatrix}$$



THANK YOU