Date: __/__/20__ Day: _____

HAND WRITTEN NOTES

CALCULAS FOR ML/AI

Written By:
Mirza Yasir Abdullah Baig

All Topics from basics to Advanced

Topics

- Fundamental Calculus Concepts
  - Differentiation
  - Partial Derivatives
  - Gradient Descent
  - Chain Rule
  - Jacobian and Hessian Matrices
- Inverse Trigonometric Functions Differentiation
- Partial Differentiation
- Higher Order Derivatives
- Optimization techniques Using Gradient Desent
- Uni-variate Optimization

Vector Calculus

- Gradient
- Divergence and Curl
- fine Integrals
- Laplacian Operator

Date: __/__/20 Day: _____

Introduction To Calculas

Calculus is a branch of mathematics that studies change. In AI and Data Science, we use calculus to understand how small changes in numbers affect the final results.

Think of it like this

- Derivatives tell us how fast something is changing.
- Integrals help us add up small pieces to find whole

Why is Calculus Important in AI and ML?

- Training Models: Machine learning models learn by adjusting numbers (called parameters). Calculus helps us figure out how to adjust them step by step.
- Optimization: We use calculus to find the "best solution", like the lowest error in predictions.
- Neural Networks: Backpropagation (the way neural networks learn) uses derivation/lives to update weights
- Data Patterns: It helps in analyzing contineous data, curves, and trends.

Simple example

Imagine you are climbing a hill

- Calculus helps you know whether you are going up or down (derivative)
- It also tells you the lowest point of the hill (minimum error), which is exactly which machine learning models try to find during training

In Short: Calculus is language of change. Without it, machine cannot learn effectively from data

Date: __/__/20__ Day: _____

Fundamentals of Differential Calculus

Differential calculus is a branch of calculus that studies derivatives and their uses.
A derivative tells us how fast a function is changing at a certain point.

It is very important in science, engineering, physics, economics and many other fields. With it, we can calculate instantaneous rates of change and the slope of curves

Key Concepts in Differential Calculus

Differential calculas deals with rates of change.
The main concept is the derivative, which measures the rate of change of a function with respect to a variable

It helps solve problems where change is not constant, and is widely used in physics, engineering, economics, biology, and more

Limits

A limits describes the behaviour of a function as its input gets closer to a certain value
Example:

$$\lim_{x \to c} f(x) = L$$

This means as x gets closer to c, the value of $f(x)$ gets closer to L.
Limits are used to define derivatives, integrals, and continuity

Date: __/__/20__ Day: _____

Continuity

A function is continuous at a point if there is no break or hole in its graphs.

For $f(x)$ to be continuous at $x = a$

- $f(a)$ exists
- Limit of $f(x)$ as $x \to a$ exists
- Limit of $f(x) = f(a)$

Derivatives

The derivatives of $f(x)$ is defined as

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

This shows the instantaneous rate of change of $f(x)$. It comes from the slope of the tangent line to the curve at point x.

Notations of Derivative

- Leibniz: dy/dx
- Lagrange: $f'(x)$ or y'
- Newton: $\dot{y}$
- Euler: $Df(x)$ or Dy

Basics Rules of Differentiation

- Product Rule

$$(uv)' = u \cdot v' + v \cdot u'$$

- Quotient Rule

$$(h/g)' = \frac{g \cdot h' - h \cdot g'}{g^2}$$

- Sum Rule:

$$(u+v)' = u' + v'$$

Date: 1/20 Day:

- Power Rule:

$$(x^n)' = n \cdot x^{n-1}$$

- Constant Multiple Rule:

$$(cf(x))' = c \cdot f'(x)$$

- Chain Rule

$$(f(g(x)))' = f'(g(x)) \cdot g'(x)$$

Differentiation of Common Functions	
Function	Derivative
Constant c	0
$\sin x$	$\cos x$
$\cos x$	$-\sin x$
$\tan x$	$\sec^2 x$
$\sin^{-1} x$	$1/\sqrt{(1-x^2)}$
$\cos^{-1} x$	$-1/\sqrt{(1-x^2)}$
$\tan^{-1} x$	$1/(1+x^2)$

Other Functions:

- Exponential : $d/dx\ (e \hat{}\ x) = e \hat{} x$, $d/dx\ (a \hat{}\ x) = a \hat{} x \ln(a)$
- Logarithmic: $d/dx(\ln x) = 1/x$, $d/dx(\log_a x) = 1/(x \ln a)$
- Polynomials: Apply the power rule to each term

First Principle of Differentiation

This method uses the definition of derivative

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

Techniques of Differentiation

- Implicit Differentiation
  - Used when y and x are mixed in one equation

Example: $x^2 + y^2 = 1 \rightarrow dy/dx = -x/y$

Date: __/__/20 Day: _____

- Logarithmic Differentiation
  - Useful for complicated products and powers.

Examples: $y = x^x \rightarrow dy/dx = x^x(\ln x + 1)$

- Parametric Differentiation

If $x = f(t)$, $y = g(t)$, then

$$dy/dx = \frac{dy/dt}{dx/dt}$$

Example: $x = \cos t$ & $y = \sin t$, $\rightarrow dy/dx = -\cot$

Applications of Differential Calculus

- Rate of change (velocity, growth etc)
- Optimization (finding max/min values)
- Curve Sketching (slopes, turning points)
- Physics (velocity) (acceleration)
- Biology (population growth, enzyme reactions)
- Engineering (stress, design, optimization)
- Finance (interest, porfolio, derivates)
- Computer Graphics (smoth curve, 3D modeling)
- Economics (marginal cost, revenue, profit)

Differentiation

Differential calculus is a part of calculus that studies rates of change. It tells us how a function changes when its input (independent variable) changes a little.

Prerequisites

Before learning differential calculus, you should know:

Date: __/__/20__ Day: _____

- Independent and dependent variables,
- Functions
- Trigonometry
- Arithmetic.

What is a limit?

For a function $y = f(x)$, the limit as $x \to a$ means: what value does the function get close to when x gets close to a?

It is written as:

$$\lim_{x \to a} f(x)$$

- A limit is unique - for a given $x \to a$, the value of $f(x)$ can't be two different things

Left-hand and Right-hand Limits

- left-hand limit: $\lim_{x \to a^-} f(x) = \lim_{h \to 0^-} f(a-h)$
- Right-hand limit: $\lim_{x \to a^+} f(x) = \lim_{h \to 0^+} f(a+h)$

Existence of Limit

$\lim_{x \to a} f(x)$ exists if:

- Both left-hand and right-hand limits exist
- Both are equal.

How to Evaluate Limits

Depending on the function, limits can be determinate (clear value) or indeterminate (uncertain forms like $0/0$, ∞/∞)

Determinate Form

If at $x = a$, $f(x)$ gives a clear value, then

$$\lim_{x \to a} f(x) = f(a)$$

Indeterminate Forms

Forms like $0/0$, ∞/∞, 0^0, 1^∞, & ∞^0

Date: __/__/20 Day: _____

Methods to solve:

- Factorization Method → cancel common Factors
- Rationalization Method → multiply by conjugate if square roots exist
- Substitution Method → put $x = a + h$, then let $h \to 0$
- When $x \to \infty$ → divide numerator and denominator by highest power of x.

L'Hospital's Rule

If the form is $0/0$ or ∞/∞:

$$\lim_{x \to a} \frac{f(x)}{g(x)} = \lim_{x \to a} \frac{f'(x)}{g'(x)}$$

Sandwich Theorem

If $f(x) \leq g(x) \leq h(x)$ and both $\lim_{x \to a} f(x) = \lim_{x \to a} h(x) = p$, then $\lim_{x \to a} g(x) = p$

Continuity, Discontinuity and Differentiability

- Continuous: at $x = a$:

$$\lim_{x \to a^-} f(x) = \lim_{x \to a^+} f(x) = f(a)$$

- Discontinuous: if $f(a)$ is not defined, or limits doesn't exist, or limit $\neq f(a)$.
- Differentiate:

$$\lim_{x \to a^-} \frac{f(x) - f(a)}{x - a} = \lim_{x \to a^+} \frac{f(x) - f(a)}{x - a}$$

Mean Value Theorem (MVT)

If $f(x)$ is continuous on $[a, b]$ and differentiable on (a, b), then

$$\Rightarrow f'(c) = \frac{f(b) - f(a)}{b - a}, \quad c \in (a, b)$$

Date: __/__/20__ Day: _____

Derivatives

- At a point:

$$\lim_{x \to c} \frac{f(x) - f(c)}{x - c}$$

- of a Function:

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

- As a rate of change:

$$\frac{dy}{dx} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x}$$

- First Principle:

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

Common Derivative Formulas

- $d/dx(c) = 0$
- $d/dx(x) = 1$
- $d/dx(x^n) = nx^{n-1}$
- $d/dx[f(g(x))] = f'(g(x)) \cdot g'(x)$
- $d/dx(\sin x) = \cos x$
- $d/dx(\cos x) = -\sin x$
- $d/dx(\tan x) = \sec^2 x$
- $d/dx(\ln x) = 1/x$
- $d/dx(e^x) = e^x$

(and so on for inverse trig function)

Other Differentiation Techniques

- Parametric differentiation
- Impliat differentiation
- Higher-order derivatives

Date: __/__/20 Day: _____

Applications of Derivatives

- Tangents and normals
- Increasing and decreasing functions
- Maxima and minima (first/second derivative test)
- Inflection points
- Solving optimization problems
- Used in physics, business, engineering, & daily life.

Differential Equations

A differential equation has an independent variable, a dependent variable, and derivatives.

Order and degree

- Order → highest derivative present
- Degree → power of highest derivative

Example: $(dy/dx)^3 + 3(d^2y/dx^2)^2$ → order 2, degree 3

Types:

- Ordinary Differential Equation (ODE)
- First-order, second-order
- Partial Differential Equation (PDE)
- Linear / Non-Linear
- Homogeneous / Non-homogeneous
- Exact and Non-Exact.
- Separable

Partial Derivatives

In machine learning, models have parameters, like weights and biases. The loss function tells us how far our predictions are from the true values. To improve the model, we need to know each parameter

Date: __/__/20__ Day: _____

affects the loss. Partial derivatives help us update each parameter separately to reduce error. This is the main idea behind gradient desent, the optimization method used to train models.

Partial Derivatives

A function of many variables has a partial derivatives with respect to one variable while keeping the other constant.

For example: For $f(x_1, x_2, \cdots x_n)$, the partial derivative with respect to x_i is written as:

$$\frac{\partial f}{\partial x_i}$$

Gradient

The gradient is a vector that points in the direction of the steepest increase of a function. In ML, it helps to know the direction to increase or decrease the loss.

Gradient Descent

Gradient descent is an optimization method that moves in the direction of steepest descent (negative gradient) to minimize the loss function

Example: linear Regression

Model:

$$f(x) = wx + b$$

w = weight

b = bias

x = input variable

Date: __/__/20 Day: _____

We calculate partial derivatives of the cost function $J(w, b)$ with respect to w and b:

$$\frac{\partial J}{\partial w} = \frac{1}{m} \sum_{i=1}^{m} (wx_i + b - y_i) x_i$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (wx_i + b - y_i)$$

where (x_i, y_i) are training samples and m is the number of samples

update Rule (Gradient Descent)

$$w = w - \alpha \frac{\partial J}{\partial w}$$

$$b = b - \alpha \frac{\partial J}{\partial b}$$

α = learning rate

Implementation In Python

```
import numpy as np
x = np.array([1, 2, 3, 4, 5])
y = np.array([100, 200, 300, 400, 500])
w = 0
b = 0
learning_rate = 0.01
epochs = 100
For epoch in range(epochs):
    predictions = w* x +b
    dw = (1/len(x)) * np.sum((predictions - y) * x)
    db = (1/ len(x))* np.sum (predictions - y)
    w -= learning_rate * dw
    b -= learning_rate * db
print("Optimal parameters: \n w = ", w, "\n b = ", b)
```

Date: __/__/20__ Day: _____

Gradient Deseent

Gradient Dècent is the main technique used to train machine learning models like linear regression, logistic regression, SVMs, and neural networks. It helps minimize the cost function by iteratively adjusting model parameters (weights and biases) to reduce the difference between predicted and actual values.

How Gradient Descent Works in Machine learning?

Training Models

In neural networks, gradient descent is used with backpropagation:

- Forward Propagation: Calculate predictions for given inputs
- Backward Propagation: Use the chain rule to compute gradients of the loss w.r.t each parameter
- Update Parameters: Gradient descent updates weights and biases layer by layer to reduce the loss

Minimizing the Cost Function

Gradient descent minimizes a loss function that measures error:

- Linear Regression: Minimizes Mean Squared Error (MSE). The update rules are:

$$w = w - \alpha \frac{\partial J(w,b)}{\partial w}, \quad b = b - \alpha \frac{\partial J(w,b)}{\partial b}$$

- Logistic Regression: Minimizes Log Loss using the sigmoid function. Updates shift the decision boundary to improve classification.

Date: __/__/20__ Day: _____

- SVMs : Minimizes Hinges Loss (and optional regularization) to find the maximum-margin hyperplane

Gradient Descent Steps In Python

```
import torch
import torch.nn as nn
import matplotlib.pyplot as plt
torch.manual_seed(42)
num_samples = 1000
x = torch.randn(num_samples, 2)
true_weights = torch.tensor([1.3, -1])
true_bias = torch.tensor([-3.5])
y = x @ true_weights.T + true_bias
class LinearRegression(nn.Module):
    def __init__(self, input_size, output_size):
        super().__init__()
        self.linear = nn.Linear(input_size, output_size)
    def forward(self, x):
        return self.linear(x)
model = LinearRegression(input_size = x.shape[1], outputsize = 1)
w = nn.Parameter(torch.rand(1, x.shape[1]))
b = nn.Parameter(torch.rand(1))
model.linear.weight = w
model.linear.bias = b
def Mean_Squared_Error(prediction, actual):
    return ((actual - prediction)**2).mean()
learning_rate = 0.001
num_epochs = 1000
for epoch in range(num_epochs):
    y_p = model(x)
    loss = Mean_Squared_Error(y_p, y)
```

Date: _/_ /20 Day: _____

```
loss.backward()
w = w - learnig_rate * w.grad
b = b - learning_rate * b.grad
model.linear.weight = nn.Parameter(w)
model.linear.bias = nn.Parameter(b)
if (epochs+1) % 100 == 0:
    print(f'Epoch [{epoch+1}/{num_epochs}],
          Loss: {loss.item():.4F}')
```

Output example shows loss decreasing as weights and bias are updated.

Learning Rate

- Too small: Slow convergence
- Too large: Overshooting or divergence
- Must choose a balanced learning rate for efficient training

Vanishing and Exploding gradients

- Vanishing gradients: Gradients get too small → early layers learn slowly
- Euplonding gradients: Gradients get too large → weight overshoot, training unstable

•Fixes

- Proper weight initialization
- ReLU activation
- Gradient clipping
- Batch normalization

Variants of Gradient Descent

- Batch GD: Uses full dataset per update (accurate but slow)
- Stochastic GD: (SGD): Use one sample per update (fast but noice)

Date: __/__/20__ Day: _____

- Mini-batch GD: Uses small batch per update (balance of speed and stability)
- Momentum: Adds past gradient info to speed up convergence.
- Nesterov Accelerated Gradient (NAG): Predicts next position using momentum for faster convergence.
- Adagrad: Adaptive learning rate per parameter based on past gradients
- RMSprop: Adaptive learning rate using moving average of squared gradients.
- Adam: Combines Momentum, RMSprop, and Adagrad for faster, stable covergence.

Advantages

- Widely used and easy to implement
- Coverages to global or good local minimum
- Scalable to large datasets and high dimensions
- Flexible with many variants.

Disadvantages

- Sensitive to learning rate and parameter initialization
- Can be slow for large datasets.
- May get stucks in local minima for non-convex func.

Conclusion

Gradient descent is the core optimization method in machine learning and deep learning. By updating weights and biases iteratively it helps models learn patterns and minimize errors. Different variants provides flexibility, efficiency and stability in training complex models.

Date: _/_ /20 Day: _____

Chain Rule

The chain Rule is a method to find the derivative of composite functions (functions inside functions) It is one of the basic and most useful rules in calculus, introduction by Gottfried Leibniz in the 17th century

Example of composite functions:

- $(3x^2 + 1)^4$
- $\sin(4x)$
- e^{3x}
- $(\ln x)^2$

Chain Rule Statement

If $f(x)$ is a composite function of $g(x)$, then:

$$d/dx [f(g(x))] = f'(g(x)) \cdot g'(x)$$

Example: $\cos(4x)$

- outer function: $f(x) = \cos(x)$
- inner function: $g(x) = 4x$

$$d/dx [\cos(4x)] = -\sin(4x) \cdot 4 = -4\sin(4x)$$

Chain Rule Theorem:

If $f = p(q(x))$, then:

$$df/dt = dp/du \cdot du/dt$$

This mean: rate of change of the outer function x rate of change of the inner function.

Date: __/__/20__ Day: _____

Steps to Apply Chain Rule (Example: $\sin(x^2)$)

- Identify if the function is composite. ($\sin(x^2)$ is composite)
- Outer function: $f(x) = \sin(x)$, Inner function: $g(x) = x^2$
- Differentiate outer: $f'(x) = \cos(x)$
- Differentiate inner: $g'(x) = 2x$
- Multiply:

$$d/dx [\sin(x^2)] = (2x)\cos(x^2)$$

Chain rule Formulas

1. $$d/dx [f(g(x))] = f'(g(x)) \cdot g'(x)$$

Example: $d/dx (\cos(2x)) = -\sin(2x) \cdot 2 = -2\sin(2x)$

2. $$dy/dx = \frac{dy}{du} \cdot \frac{du}{dx}$$

Example: Let $y = \cos(2x)$, $u = 2x$

$$dy/dx = \frac{dy}{du} \cdot \frac{du}{dx} = -\sin(u) \cdot 2 = -2\sin(2x)$$

Proof of Chain Rule

For $y = f(u(x))$

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

using limit defination of derivatives, we arrive at:

$$\frac{dy}{dx} = f'(g(x)) \cdot g'(x)$$

Double Chain Rule

If a function is nested multiple times, use double chain rule:

For $f(x) = p(q(r(x)))$

$$\frac{df}{dx} = \frac{df}{dp} \cdot \frac{dp}{dq} \cdot \frac{dq}{dr} \cdot \frac{dr}{dx}$$

Date: _/_ /20 Day: _____

Example: $y = (\sin(2x))^2$

$y' = 2(\sin(2x)) \cdot (\cos(2x)) \cdot 2 = 4\sin(2x)\cos(2x)$

Chain Rule for partial Derivatives

For multivariable functions, the chain rule uses the Jacobian matrix.

If $y = f(u)$ where $u = g(x)$

$$\frac{\partial(y_1, \ldots, y_k)}{\partial x_i} = \frac{\partial(y_1, \ldots, y_k)}{\partial(u_1, \ldots, u_m)} \cdot \frac{\partial(u_1, \ldots u_m)}{\partial x_i}$$

Applications of Chain Rule

- Finding how pressure changes with time
- Rate of change of molecular speed.
- Checking if a function is increasing or decreasing
- Rate of change of distance b/w moving objects

Summary

- The Chain Rule is used to differentiate composite functions. It applies in single-variable calculus, double-nested functions, and even partial derivatives. It is widely used in math, physics, and machine Learning.

Jacobian & Hessian Matrices

In multivariate optimization, we deal with functions that depend on many variables

$$z = f(x_1, x_2, x_3, \ldots, x_n)$$

Here, $x_1, x_2, \ldots, x_n$ are the decision variables. The goals is to find values of these variables that optimizes (maximum or minimize) the function z

Date: __/__/20 Day: _____

For one variable (univariate optimization), it's easy to visualize

- x-axis → input value
- y-axis → function value

But for multivariate optimization, we often need 3D graphs. If variables > 2, visualization becomes very hard.

Gradient

In ~~universe~~ univariate optimization, the necessary condition for a minimum is

$$f'(x) = 0$$

In multivariable optimization, the same idea extends to the gradient.

- The gradient is a vector of partial derivatives
- Each component tells how the function changes with respect to one variable

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n} \right]$$

Note: The gradient at a point is always perpendicular (orthogonal) to the contour lines of the function

Hessian

In the univariate case, the second derivative tells us if a point x a minimum:

$$f''(x) > 0 \Rightarrow \text{minimum}$$

For multivariate functions, we use the Hessian matrix instead of just $f''(x)$

Date: __/__/20__ Day: _____

- The Hessian is a square matrix (n×n) containing all the second-order partial derivatives

$$\nabla^2 f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Notes on Hessian

- The Hessian is always a symmetric matrix
- If all eigenvalues of the Hessian are positive, the matrix is positive definite, meaning the function has a minimum at that point

In Short

- Gradient → generalization of first derivative (slope = 0)
- Hessian → generalization of second derivative (min or max)

Inverse Trigonomatric Functions Differentiate

The derivative of an inverse trigonometric function shows how fast the function changes with respect to its variable.

To find these derivatives, we usually:

- Write the inverse trig function as normal trig function
- Differentiate both sides using implicit differentiation
- Simplify using trig identities

Inverse Trigonometric Functions

Inverse trig functions are the opposite of sine,

Date: __/__/20__ Day: _____

Partial Differentiation

A partial derivative is the derivative of a function with more than one variable, where we differentiate with respect to only one variable at a time, treating other as constants

Example

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

- with respect to x: $\partial f / \partial x = 2x$
- with respect to y: $\partial f / \partial y = 3y^2$

Symbol: ∂ (different from d used in ordinary derivatives)

Formula:

For a function $f(x, y, z)$:

$$f_x = \partial f / \partial x = \lim_{h \to 0} \frac{f(x+h, y, z) - f(x, y, z)}{h}$$

$$f_y = \partial f / \partial y = \lim_{h \to 0} \frac{f(x, y+h, z) - f(x, y, z)}{h}$$

$$f_z = \partial f / \partial z = \lim_{h \to 0} \frac{f(x, y, z+h) - f(x, y, z)}{h}$$

Different Orders of Partial Derivatives

- First order

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

$f_x = 2xy, \quad f_y = x^2 + 6y$

At $(2, 1)$: $f_x = 4, \quad f_y = 10$

- Second Order

$$f_{xx} = \frac{\partial^2 f}{\partial x^2}, \quad f_{yy} = \frac{\partial^2 f}{\partial y^2}, \quad f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$$

Date: _/_/20 Day: _____

For same example:

- $f_{xx} = 2y \Rightarrow 2$ at $(2,1)$
- $f_{yy} = 6$
- $f_{xy} = 2x \Rightarrow 4$ at $(2,1)$

Rules of Partial Differentiation

- Product Rule

$$v = f(x,y) \cdot g(x,y) \Rightarrow v_x = f_x g + f g_x$$

- Quotient Rule:

$$v = \frac{f(x,y)}{g(x,y)} \Rightarrow v_x = \frac{f_x g - f g_x}{g^2}$$

- Power Rule:

$$v = (f(x,y))^n \Rightarrow v_x = n(f(x,y))^{n-1} \cdot f_x$$

- Chain Rule:

- If $z = f(x,y)$, $x = g(t)$, $y = h(t)$

$$\frac{\partial z}{\partial t} = \frac{\partial z}{\partial x}\frac{dx}{dt} + \frac{\partial z}{\partial y}\frac{dy}{dt}$$

- If $x = g(u,v)$, $y = h(u,v)$

$$\frac{\partial z}{\partial v} = \frac{\partial z}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial z}{\partial y}\frac{\partial y}{\partial v}$$

Total Derivative Vs Partial Derivative

Feature	Partial Derivative	Total Derivative
Symbol	∂	d
Meaning	change wrt one variable, keeping others constant	change wrt one variable considering all variables dependence.
Example	for $f(x,y) = x^2y$: $\partial f/\partial x = 2xy$	Total derivatives: $df/dt = 2xy \cdot dx/dt + x^2 \cdot dy/dt$

Date: __/__/20__ Day: _____

Applications of Partial Derivatives

- Math Models → Simplify complex equations
- Engineering → used in Control systems & dynamics
- Chemistry → Study reaction rates (Kinetics)
- Biology → analyze population change wrt birth, deadth, migration
- Economics / Finance → Predict outcome, optimize profit based on inputs like labor & capital.

Summary

- Partial derivatives study how a function changes wrt one variable at a time.
- We have first, Second, and mixed partials.
- Rules: product, quotient, power, chain.
- Used widely in math, science, engineering, economics.

Higher Order derivatives

A higher order derivative means taking the derivative of a function again and again.

- The first derivative $f'(x)$: shows the slope or rate of change
- The second derivative $f''(x)$: shows curvature/ concavity (helps to find maximum, minimum)
- The third derivative $f'''(x)$: derviative of 2nd derivative.
- In general, the nth derivative $f^n(x)$: Keeps showing the "rate of change of the rate of change" as you go higher.

Second Order Derivatives

The second derivative tells how the slope itself is changing:

Date: _/_/20

Day: _____

If $y = f(x)$

- first derivative: $f'(x) = dy/dx$
- Second derivative: $f''(x) = d^2y/dx^2$

It is useful for finding maxima, and optimal points of functions.

Example 1

$$y = \frac{1}{x^2+1}$$

Using, quotient rule,

$$y' = \frac{1-x^2}{(x^2+1)^2}$$

Differentiate again given:

$$y'' = \frac{(x^2+1)^2(-2x) - (1-x^2)(2(x^2+1)(2x))}{(x^2+1)^4}$$

At $x = 1$, $y'' = 0$

So, the second derivative at $x = 1$ is 0

Third Order Derivatives

The third derivative is simply the derivative of the second derivative

- If $f(x) \rightarrow$ first derivative $f'(x)$
- Then $f''(x) \rightarrow$ second derivative
- Then $f'''(x) \rightarrow d^3y/dx^3 \rightarrow$ third derivative

Example 2:

$y(x) = 3x^2 + 6x + 4$

- $y'(x) = 3x^3\ 9x^2 + 12$
- $y''(x) = 18x$
- $y'''(x) = 18$

At $x = 1$, $y'''(1) = 18$

Date: __/__/20__ Day: _____

Higher Order Derivatives in Parametric Form

If both x and y are given as functions of a parameter t:

- First derivative: $dy/dx = (dy/dt) / (dx/dt)$
- Second derivative:

$$d^2y/dx^2 = d/dt \left(\frac{dy/dt}{dx/dt} \right) \times \frac{dt}{dx}$$

This, way, we can compute higher derivatives in parametric equations.

Applications of Higher Order Derivatives.

They are used in:

- Finding acceleration from displacement-time functions
- Checking maxima and minima.
- Understanding the shape of graphs (concavity)
- Second derivative test in optimization.

Example 3: $f(x) = x^3$

- $f'(x) = 3x^2$, $f''(x) = 6x$, $f'''(x) = 6$

Example 4: $f(x) = ex + \sin(x)$

- $f'(x) = e^x + \cos(x)$
- $f''(x) = e^x - \sin(x)$
- $f'''(x) = e^x - \cos(x)$

At $x = 0$, $f'''(0) = 1 - 1 = 0$

Example 5: $f(x) = e^x \sin(x)$

- $f'(x) = e^x (\sin(x) + \cos(x))$
- $f''(x) = e^x (2 \cos(x))$

At $x = 0$, $f''(0) = 2$

Date: _/_ /20

Day: _____

Optimization Techniques using Gradient Descent

Gradient Descent is one of the most popular algorithms used to optimize machine learning models. It works by taking steps in the direction of the slope (graddient) to reach the minimum error. However, there are several techniques to make it faster and more effective.

- Learning Rate Scheduling
  - The learning rate controls the step size of Gradcent Desent
  - Instead of keeping it fixed, we can change it during training
  - Usually, the learning rate decreases after some iterations
  - This helps the algorithm move faster at the start and avoid overshooting near the minimom

- Momentum - Based Updates
- Momentum adds a part of previous update to the current one.
- This helps the algorithm roll over small bumps (local minima) and move faster towards the best solution:

Formula:

$$v = \beta V + (1 - \beta) dW$$

$$W = W - \alpha V$$

Here:

V = velocity (like in Physics)

dW = gradient

α = learning rate

β = momentum (usually 0.9)

Date: __/__/20 Day: _____

- Batch Normalization
  - Normalizes the input to each layer of a neural network
  - Prevents gradients from becoming too small (vanishing) or too large (exploding)
  - Helps the model train faster and more stably.

- Weight Decay
  - A regularization method.
  - Adds a small penalty to large weights in the cost function
  - Prevents overfitting and improves generalization

- Adaptive learning Rates:
  - Instead of using a fixed learning rate, these methods adjust it automatically
  - Popular Algorithms
    - Adagrad
    - RMSprop
    - Adam
  - They use past gradient information to decide the best step size, making training faster and more accurate.

- Second-Order Methods
  - These methods use second derivatives (curvature) (information) of the cost function.
  - Example: Newton's Method, Quasi-Network Methods
  - They converge faster but are computationally expensive and sometimes less stable.

Date: __/__/20__ Day: _____

Extensions of Gradient Descent

- Momentum Method (Physics Analogy: ball rolling downhill)
  - Uses the average of past gradients to speed up training
  - Cancels out noisy updates and focuses on the main direction

RMSprop (by Geoffry Hinton)

- Keeps a moving average of squared gradients
- Updates weights using:

$$S = \beta S + (1 - \beta) dW^2$$

$$W = W - \alpha \, dW / \sqrt{(S + \varepsilon)}$$

Adam optimization (Most popular)

- Combines Momentum + RMSprop + bias correction
- Recommended hyperparameters
  - α = 0.001
  - β = 0.9
  - $\beta 2$ = 0.999
  - ε = 10^-8

Formula:

$$V = \beta_1 V + (1 - \beta_1) dW$$

$$S = \beta_2 S + (1 - \beta_2) dW^2$$

$$V = V / (1 - \beta_1 ^ 1)$$

$$S = S / (1 - \beta_2 ^ i)$$

$$W = W - \alpha V / \sqrt{(S + \varepsilon)}$$

Summary: This can be improved using techniques like learning rate scheduling, momentum batch normalization, weight decay, adaptive methods and second-order methods. Adam is most widely used today, it combines benefits of other.

Date: __/__/20__ Day: _____

Vector Calculus (Gradient)

The gradient is a key idea in calculus. It is like a derivative, but for functions with many variables. It shows the direction of fastest increase of a function. Gradients are very useful in calculus, optimization, physics and machine learning

Mathematical Definition

For a scalar function:

$$f(x_1, x_2, \cdots, x_n)$$

The gradient is a vector of partial derivatives:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n} \right)$$

Each part show how f changes with one variable.

Gradient in 2D and 3D

- For $f(x, y)$:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$$

- for $f(x, y, z)$:

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)$$

Geomatric Meaning:

- Direction of fastest increase → gradient points to where the function grows most quickly.
- Size of gradient → shows how steep the growth is
- Perpendicular to level curves → gradient is always at right angles to curves (or surfaces) where function is constant.

Example

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

Date: __/__/20__ Day: _____

- Global optimum: The best solution overall
- Local optimum: Best among nearby values, but not the absolute best.
- Derivative: Rate of change of f(x)
- Gradient: In univariate case, same as derivative
- Critical points: Where $f'(x) = 0$ or undefined
- Convexity: line b/w two points lies above the func.
- Concavity: line b/w two points lies below the function

Steps in Univariate Optimization

- Define the objective function f(x)
- Decide if we want to minimize or maximize.
- Check if there are constraints
- Choose an optimization algorithm
- Run the algo and set convergence criteria
- Validate results and tune hyperparameters if required.

Necessary & Sufficient Conditions

For a point x to be an optimizer

- First-order condition: $f'(x) = 0$
- Second-order condition for minimum: $f''(x) > 0$
- Second-order condition for maximum: $f''(x) < 0$

Example

$f(x) = 3x^4 - 4x^3 - 12x^2 + 3$

First derivative

$f'(x) = 12x^3 - 12x^2 - 24x$

- $12x(x^2 - x - 2)$

Critical points → Solve

$x = 0, \; x = -1, \; x = 2$

Date: __/__/20 Day: _____

Second derivative

$f''(x) = 36x^2 - 24x - 24$

Test Critical points

- $f''(0) = -24 \rightarrow$ Not minimum
- $f''(-1) = 36 > 0 \rightarrow$ Minimum
- $f''(2) = 72 > 0 \rightarrow$ Minimum

So, minimizers are

$f(-1) = -2$

$f(2) = -29$

Python

```
import numpy as np
from scipy.optimize import minimize_scalar
def objective_function(x):
    return x**2 + 3*x + 2
result = minimize_scalar(objective_function)
print("Optimal Value:", result.x)
print("Optimal function Value:", result.fun)
```

Output

Optimal Value: -1.5

Optimal function value: -0.25

Summary

Univariate Optimization helps find the best value of a function with one variable. It involves defining the function, finding critical points, checking conditions, and using algorithms (like scipy).

Date: __/__/20

Day: _____

Uni-variate Optimization

Optimization is important in data science because it helps us find the best parameters for a machine learning model to minimize loss. One common approach is gradient-based methods.

What is Univariate Optimization?

Univariate optimization means finding the best value of a function with respect to one variable (keeping others fixed)

Example: maximizing profit, minimizing distance, or finding the smallest/largest value of a function.

Mathematical form:

minimize $f(x)$, with respect to x

subject to $a \leq x \leq b$

- $f(x)$: objective function
- x: Decision variable
- $a \leq x \leq b$: Constraint

Types of Optimization

- Constrained optimization: Has conditions (constraints) that solutions must satisfy.
- Unconstrained optimization: No restrictions.

Key Terms

- Objective Function: ($f(x)$) → The function we optimize
- Decision variable (x): The variable we change to optimize
- Feasible region: Range of x values allowed by constraints

Date: __/__/20 Day: _____

Step 1: Find gradient

$$\frac{\partial f}{\partial x} = 2x, \quad \frac{\partial f}{\partial y} = 6y$$

So,

$$\nabla f = (2x, 6y)$$

Step 2: At (1, 2)

$$\nabla f(1, 2) = (2, 12)$$

This means the function grows fatest in direction (2, 12)

Python Example

- Symbolic Gradient (sympy)

```
import sympy as sp
x, y = sp.symbols('x, y')
f = x**2 + 3*y**2
grad_f = [sp.diff(f, var) for var in (x,y)]
print("Gradient:", grad_f)
```

Evaluate at (1, 2)

```
grad_f_func = [sp.lambdify((x,y), expr) for expr in grad_f]
grad_value = [fun(1, 2) for func in grad_f_func]
print("Gradient at (1, 2):", grad_value)
```

Visualize Gradient Field

```
import numpy as np
import matplotlib.pyplot as plt
X, Y = np.meshgrid(np.linspace(-3, 3, 10), np.linspace(-3,3,10))
U, V = 2*X, 6*Y
plt.figure(figsize=(6, 6))
plt.quiver(X, Y, U, V, color='r', angles='xy').
plt.xlabel('x'); plt.ylabel('Y')
plt.title('Gradient field of f(x,y) = x² + 3y²")
```

Date: __/__/20__ Day: _____

Applications of Gradient

- Optimization (Gradient Descent)

Used in ML to minimize loss:

$$\theta \leftarrow \theta - \alpha \nabla f(\theta)$$

where θ = parameters, α = learning rate.

- Physics
  - Electric field: $E = -\nabla V$
  - Gravity: $F = -\nabla U$

- Computer Vision

Image gradients (like Sobel filters) help find edges.

- Robotics

Used for path planning → robots move along gradient directions to reach goals.

Divergence and Curl

These are operators in vector calculus

- Divergence → gives a scalar (number). It shows how much a vector field spreads out (source) or squeezes in (sink).
- Curl → gives a vector. It shows the rotation or circulation of vector field.

What is Divergence?

Divergence measures how a vector field spreads (diverges) or converges (compresses) at a point.

For a vector field $F = (F_1, F_2, F_3)$

$$\text{div } F = \nabla \cdot F = \frac{\partial F_1}{\partial x} + \frac{\partial F_2}{\partial y} + \frac{\partial F_3}{\partial z}$$

Date: __/__/20__ Day: _____

Result is a scalar.

Positive → source, Negative → sink.

What is Curl?

Curl measures how a vector field rotates around a point

For $F = (F_1, F_2, F_3)$

$$curlF = \nabla \times F = \left(\frac{\partial F_3}{\partial y} - \frac{\partial F_2}{\partial z}, \frac{\partial F_1}{\partial z} - \frac{\partial F_3}{\partial x}, \frac{\partial F_3}{\partial x} - \frac{\partial F_1}{\partial y}\right)$$

Result is a vector

Shows circulation or twisting of field line.

Divergence of Curl

For any smooth vector field F:

$$\nabla \cdot (\nabla \times F) = 0$$

This means the divergence of curl is always zero

This is proven using vector calculus identities and mixed partial derivatives (Clairaut's theorem)

Line Integrals

A line Integral is used to calculate a function along a curve or path. It is important in Pysics, engineering and maths.

- It helps measure total effects (like work, flux, heat) along a path.
- It is called path integral, curve integral, or curvilinear integral
- Common Use: Finding work done by a force when moving an object along a curve.

Date: __/__/20__ Day: _____

Defination

A line Integral is the integral of a function along a line or curve. It is likely summing up the function's values over all small parts of the curve

Formula of Line Integral

For Scalar Fields

If f is scalar field and $r(t)$ is the parametric curve

$$\int_C f(r(t)) \, |r'(t)| \, dt$$

- f = scalar field
- $r(t)$ = parametric curve
- $|r'(t)|$ = magnitude of derivative of $r(t)$
- Limits a, b = start and end of curve

For Vector Fields

If F is a vector field and $r(t)$ is curve

$$\int_C F \cdot dr = \int_a^b F(r(t)) \cdot r'(t) \, dt$$

Dot product is used because both are vector

Differential Form of Line Integral

If $F = (P, Q, R)$

$$\int_C F \cdot dr = \int_C (P dx + Q dy + R dz)$$

Here, $dr = (dx, dy, dz)$

Date: __/__/20__ Day: _____

Steps to Evaluate line Integral

- Write the parametric form of curve $r(t)$
- Find differentials, dx, dy, dz
- Choose correct formula (scalar or vector field)
- Substitute parametric equations into formula
- Multiply (scalar case) or take dot product (vector)
- Integrate with in given limits

Theorem of Line Integral

$$\int_a^b F'(x)\,dx = F(b) - F(a)$$

It connects line integrals with antiderivatives

Applications of Line Integral

- Work done by gravitational or other forces
- Work done by force on moving object
- Rate of chemical reaction (Law of Mass Action)
- Magnetic field around a conductor
- Voltage induced in a loop
- Many uses in engineering & physics.

Laplace Operator

The Laplace operator (Δ or ∇^2) is a second-order differential operator used in physics, engineering, and machine learning. It is defined as divergence of gradient of a scalar field.

For a Function $f(x, y, z)$

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}$$

It measures how much the value of a function at a

Date: __/__/20__ Day: _____

point differs from the values nearby.

Interpretation of Laplacian.

- Physical interpretation
  - Shows how far a function is form its average value at a point.
  - Used in ML for:
    - Smoothing noisy data (diffusion)
    - Improving feature representation
    - Revealing hidden structures in datasets

- Geometric Interpretation
  - Laplacian describes the curvature of a function at a point.
    - If $\nabla^2 f > 0 \rightarrow$ func curves up (local minimum)
    - If $\nabla^2 f < 0 \rightarrow$ func curves down (local maximum)
    - If $\nabla^2 f = 0 \rightarrow$ func is balanced (steady state)
- Important in manifold learning and graph ML.

Mathematical Formula

$$\nabla^2(af + bg) = a\nabla^2 f + b\nabla^2 g$$

Applications

- Graph-Based Learning (clustering)
- Laplacian Eigenmaps (Dimensionality Reduction)
- Regularization in ML
- Neural Networks & DL (CNNs, KNN)