Machine Learning

Instructor: Chandra Sekhar V

Consider the quadratic function $f(t) = 3t^2 - 2t + 1$:

- **Domain**: Since $f(t)$ is a polynomial, there are no restrictions on the input values t. Thus, the domain is all real numbers $(-\infty, \infty)$.
- **Range**: The range depends on the vertex of the parabola. This function opens upwards (as the coefficient of t^2 is positive), so the range is $[f(t_{\text{vertex}}), \infty)$, where $t_{\text{vertex}} = -b/(2a)$.

2. Function Composition

- **Basics**:
  - Composition combines two functions $f(x)$ and $g(x)$: $(f \circ g)(x) = f(g(x))$ or $(g \circ f)(x) = g(f(x))$.
- **Deeper Concepts**:
  - **Order matters** in composition: $(f \circ g)(x) \neq (g \circ f)(x)$ in general.
  - Domain considerations are critical. For example, if $g(x)$ outputs a value outside $f(x)'s$ domain, the composition is undefined.

Function Composition Example

Let's consider two functions:

1. $f(x) = x^2 + 1$: Squares the input and adds 1.
2. $g(x) = \sqrt{x}$: Computes the square root of the input (only defined for $x \geq 0$).

Compositions:

1. $(f \circ g)(x) = f(g(x)) = f(\sqrt{x}) = (\sqrt{x})^2 + 1 = x + 1$, defined for $x \geq 0$.
2. $(g \circ f)(x) = g(f(x)) = g(x^2 + 1) = \sqrt{x^2 + 1}$, defined for all x since $x^2 + 1 \geq 0$.

Key Points:

- The **domain** of $f \circ g(x)$ is restricted to $x \geq 0$ because $g(x)$ (square root) is undefined for $x < 0$.
- The **domain** of $g \circ f(x)$ is all real numbers because $x^2 + 1$ is always non-negative.

The Derivative: A Deeper Explanation

The derivative of a function $f(x)$ is a fundamental concept in calculus that describes how the function $f(x)$ changes as x changes. It has both **conceptual** and **geometrical** interpretations that make it incredibly useful in mathematics, physics, engineering, and beyond.

1. Formal Definition

The derivative of $f(x)$ at a point x is defined as:

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

- **Numerator**: $f(x+h) - f(x)$ measures the change in the output of the function as x increases by a small amount h.
- **Denominator**: Dividing by h scales this change per unit increase in x, giving the rate of change.

Intuition:

- The derivative calculates the **instantaneous rate of change** of $f(x)$ at a specific x.
- It approximates the slope of the **tangent line** to the curve $y = f(x)$ at that point.

Role of the Denominator h:

1. **Represents the Change in Input (Step Size):**
   - The denominator h measures how much the input x is incremented.
   - For example, if $h = 1$, we are calculating the average rate of change over an interval of width 1.
2. **Scaling the Change in Output:**
   - The numerator $f(x+h) - f(x)$ gives the **change in the function's output** over the interval.
   - Dividing by h gives the **rate of change per unit of x**. It answers the question: "How much does $f(x)$ change for every unit increase in x?"

   Example:

   - If $h = 2$ and $f(x+h) - f(x) = 6$, then the rate of change is $\frac{6}{2} = 3$, meaning $f(x)$ increases by 3 units for every unit increase in x.

Numerical Example

Consider $f(x) = x^2$ at $x = 1$. Let's compute $\frac{f(1+h)-f(1)}{h}$ for decreasing values of h:

1. **Formula:**

$$\frac{f(1+h)-f(1)}{h} = \frac{(1+h)^2 - 1^2}{h} = \frac{1+2h+h^2-1}{h} = 2+h.$$

2. **Values for Different h:**
   - $h = 1$: $\frac{f(1+1)-f(1)}{1} = 2+1 = 3.$
   - $h = 0.1$: $\frac{f(1+0.1)-f(1)}{0.1} = 2+0.1 = 2.1.$
   - $h = 0.01$: $\frac{f(1+0.01)-f(1)}{0.01} = 2+0.01 = 2.01.$
   - $h = 0.001$: $\frac{f(1+0.001)-f(1)}{0.001} = 2+0.001 = 2.001.$

3. **As $h \to 0$:**
   - The slope approaches 2, the value of the derivative $f'(1) = 2$.

1. **Tangent Line Approximations:**
   - Tangent lines provide linear approximations to non-linear functions near a specific point.
   - For small changes around $x = a$, $f(x) \approx f'(a)(x - a) + f(a)$.
2. **Generalization:**
   - This method works for any differentiable function $f(x)$ at any point $x = a$.
   - The steeper the curve at a, the larger the slope $f'(a)$.

The function is: $f(x) = \sqrt{x}$

2. Tangent Line Equation

The tangent line to $f(x)$ at $x = 4$ is given by the formula:

$$y = f'(4)(x-4) + f(4)$$

Here:

- $f'(4)$: Slope of the tangent line at $x = 4$.
- $f(4)$: Value of the function at $x = 4$.

3. Calculate $f(4)$ and $f'(4)$

- $f(4) = \sqrt{4} = 2$
- $f'(4) = \frac{1}{2\sqrt{4}} = \frac{1}{4}$

4. Write the Tangent Line Equation

Substitute these values into the tangent line equation:

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

$$y = \frac{1}{4}x - 1 + 2$$

$$y = \frac{1}{4}x + 1$$

5. Approximation for $x = 4.1$

To estimate $\sqrt{4.1}$, substitute $x = 4.1$ into the tangent line equation:

$$y = \frac{1}{4}(4.1) + 1$$

$$y = 1.025 + 1 = 2.025$$

Thus, $\sqrt{4.1} \approx 2.025$.

Tangent Line Applications in Machine Learning:

The concept of tangent lines is fundamental in machine learning, particularly when working with optimization algorithms like gradient descent.

In the context of gradient descent and optimization, tangent lines play a crucial role in providing guidance, magnitude, and efficiency for parameter updates.

1. Gradient Descent: Optimization in Machine Learning Overview:

- In machine learning, the goal is often to minimize a loss function $L(w)$, which measures how well a model's predictions match the true values.
- Tangent lines are essential for calculating the slope of the loss function at any point
- The slope (or gradient) is used to adjust the model's parameters to reduce the loss.

How Tangent Lines Are Used

- The slope of the tangent line to $L(w)$ at a specific point indicates the direction and rate of change of $L(w)$.
- Gradient descent updates parameters w iteratively:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot L'(w)$$

  Here:

  - $L'(w)$: Slope of the tangent line (derivative of $L(w)$).
  - η: Learning rate (step size).

1. **Goal**:
    - The aim in machine learning is to minimize a **loss function** $L(w)$, which quantifies how far off a model's predictions are from the actual values.
    - w: Represents model parameters (e.g., weights).
2. **Role of Tangent Lines**:
    - The tangent line to the loss function $L(w)$ at a point gives the **slope** or **gradient** ($L'(w)$).
    - This slope indicates:
        - **Direction**: Should we increase or decrease w to minimize $L(w)$?
        - **Magnitude**: How much should w change to effectively reduce $L(w)$?
3. **Update Rule**:
    - In **gradient descent**, the model's parameters w are updated iteratively:

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot L'(w_{\text{old}})$$

    - Here:
        - η: Learning rate (controls step size).
        - $L'(w)$: Slope of the loss function (from the tangent line).
4. **Why It Works**:
    - The slope (gradient) tells us how steep the curve is and helps us move toward the direction of the minimum w.

1. Guidance

- **What it means:**
  - The slope of the tangent line at a point w (computed as $L'(w)$) tells us whether to **increase** or **decrease** w to minimize the loss function $L(w)$.
- **How it works:**
  - If $L'(w) > 0$: The tangent line slopes **upward**, meaning w is too large, and we need to **decrease** w.
  - If $L'(w) < 0$: The tangent line slopes **downward**, meaning w is too small, and we need to **increase** w.
  - If $L'(w) = 0$: The slope is flat, meaning we are at the minimum, and no further updates are needed.

  - **Example**: For the loss function $L(w) = (w-2)^2$:
    - At $w = 0$: $L'(w) = 2(0-2) = -4$ → Negative slope → Increase w.
    - At $w = 3$: $L'(w) = 2(3-2) = 2$ → Positive slope → Decrease w.

- **Deeper Concepts:**
  - Rules of differentiation:
    - **Power Rule:** $\frac{d}{dx}(x^n) = nx^{n-1}$.
    - **Chain Rule:** $\frac{d}{dx}[g(f(x))] = g'(f(x))f'(x)$.
    - **Product Rule:** $(uv)' = u'v + uv'$.
    - **Quotient Rule:** $\left(\frac{u}{v}\right)' = \frac{u'v - uv'}{v^2}$.

- **Norms of Vectors:**

- In machine learning, vector norms are used to measure the magnitude (or length) of a vector.

- Norms are crucial for optimization, regularization, distance measurement, and assessing the scale of vectors.

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

- **Definition:**
  - The L_1 norm is the **sum of the absolute values** of all elements in the vector.
  - For a vector $x = [x_1, x_2, ..., x_n]$:

$$\|x\|_1 = |x_1| + |x_2| + ... + |x_n|$$

- **Geometric Interpretation:**
  - It represents the "Manhattan distance" (or taxicab geometry) between the vector and the origin in the vector space.

L1 norm

- **Machine Learning Applications:**
  1. **Feature Sparsity:**
     - L_1-based regularization (e.g., Lasso regression) promotes sparsity in the model by driving some coefficients to zero, effectively selecting features.
  2. **Robustness:**
     - The L_1 norm is robust to outliers, as it minimizes absolute differences instead of squared differences.
- **Example:**
  - Given $x = [1, -2, 3]$:

$$\|x\|_1 = |1| + |-2| + |3| = 1 + 2 + 3 = 6$$

L2 norm

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

- **Definition:**
  - The L_2 norm is the **Euclidean norm**, representing the **straight-line distance** between the vector and the origin.
  - For a vector $x = [x_1, x_2, ..., x_n]$:

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$$

- **Geometric Interpretation:**
  - It measures the length (or magnitude) of the vector in the Euclidean space.
- **Machine Learning Applications:**
  1. **Smoothness:**
     - L_2-based regularization (e.g., Ridge regression) penalizes large coefficients without driving them to zero, ensuring smooth and less complex models.
  2. **Gradient Descent:**
     - The L_2 norm is commonly used to compute distances and gradients in optimization problems.

       - **Example:**
         - Given $x = [1, -2, 3]$:

$$\|x\|_2 = \sqrt{1^2 + (-2)^2 + 3^2} = \sqrt{1 + 4 + 9} = \sqrt{14} \approx 3.74$$

L-Infinity norm

$$\|x\|_\infty = \max(|x_i|)$$

- **Definition:**
  - The L_∞ norm is the **maximum absolute value** among the elements of the vector.
  - For a vector $x = [x_1, x_2, ..., x_n]$:

$$\|x\|_\infty = \max(|x_1|, |x_2|, ..., |x_n|)$$

- **Geometric Interpretation:**
  - It represents the distance in "Chebyshev geometry," where the metric considers the largest coordinate displacement.

  - **Example:**
    - Given $x = [1, -2, 3]$:

$$\|x\|_\infty = \max(|1|, |-2|, |3|) = 3$$

Matrices

Topic	Machine Learning Applications
• Matrix Multiplication	Neural networks (forward propagation), data transformation
• Matrix Operations	Covariance matrices, solving systems of equations
• Dot Product	Cosine similarity, weighted sums in neural networks
• Orthogonal Vectors	PCA, feature independence
• Gaussian Elimination	Solving linear regression problems
• Linear Dependence	Feature selection, eliminating redundant features
• Rank of a Matrix	Dimensionality reduction, low-rank approximations

Why Orthogonality Matters in PCA

- **Redundancy Removal:**
- Orthogonal components ensure no redundancy (correlation) in the transformed data.
- **Feature Selection:**
- The first few components explain most of the variance, allowing us to reduce dimensions while retaining information.
- **Model Stability:**
- Using orthogonal features improves numerical stability in machine learning algorithms.

- Orthogonal vectors ensure that features or components are independent and contribute unique information.
- In advanced scenarios like PCA, orthogonality simplifies the data representation, removes redundancy, and improves the efficiency of machine learning models.
- This approach is widely used in dimensionality reduction, feature engineering, and unsupervised learning tasks.

Orthogonal Vectors

Definition:

Two vectors $\mathbf{a}$ and $\mathbf{b}$ are **orthogonal** if their dot product is zero:

$$\mathbf{a} \cdot \mathbf{b} = 0$$

Geometrically, this means that the two vectors are **perpendicular** to each other (their angle is 90°).

Importance of Orthogonal Vectors

1. Feature Independence in Machine Learning:

Interpretation:

- If two feature vectors (columns in a dataset) are orthogonal, they are completely independent of each other.
- This ensures that one feature does not contribute redundant information, leading to better models and easier interpretation.

Applications:

- Orthogonality ensures no correlation between features.
- Orthogonal vectors simplify computations in linear models and reduce multicollinearity.

Why Keep Both Features?
Unique Contribution:

- Orthogonal features are completely independent, meaning they describe different aspects of the data.
- Removing one feature would result in a loss of information.

Dataset Example:

Suppose we have two features:

1. Feature $x_1 = [1, 0]$ (e.g., horizontal movement).
2. Feature $x_2 = [0, 1]$ (e.g., vertical movement).

These features are orthogonal, meaning they are independent:

- $x_1 \cdot x_2 = 0.$

Retaining Both Features:

- x_1: Captures information about horizontal variability.
- x_2: Captures information about vertical variability.
- If you remove one feature, you lose the ability to describe one axis entirely, reducing the descriptive power of your model.

Gaussian Elimination

- Gaussian elimination is a systematic method for solving systems of linear equations.
- It transforms a given system into an equivalent triangular form (row echelon form) using elementary row operations.
- Once in this form, the solution can be easily obtained by back-substitution

Key Steps in Gaussian Elimination

1. **Augmented Matrix Formation**:
   - Write the system of equations in matrix form:
     $$A\mathbf{x} = \mathbf{b}$$
     Combine A (the coefficient matrix) and $\mathbf{b}$ (the right-hand side vector) into an **augmented matrix**:
     $$[A|\mathbf{b}]$$
2. **Forward Elimination**:
   - Eliminate variables below the pivot (diagonal) element by performing row operations:
     - **Row Swapping**: Swap rows to ensure a non-zero pivot.
     - **Scaling**: Multiply a row by a scalar.
     - **Row Replacement**: Replace a row by subtracting a multiple of another row.
3. **Back Substitution**:
   - Once the augmented matrix is in **row echelon form** (upper triangular matrix), solve for each variable starting from the last equation.

System of Linear Equations:
$$\begin{aligned} 2x + y - z &= 8 \\ -3x - y + 2z &= -11 \\ -2x + y + 2z &= -3 \end{aligned}$$

Step 1: Augmented Matrix

Write the system as an augmented matrix: $\begin{bmatrix} 2 & 1 & -1 & 8 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{bmatrix}$

Step 2: Forward Elimination

1. **First Pivot (Row 1):**
   - Divide the first row by 2 to make the pivot element 1:

$$\begin{bmatrix} 1 & 0.5 & -0.5 & 4 \\ -3 & -1 & 2 & -11 \\ -2 & 1 & 2 & -3 \end{bmatrix}$$

   - Eliminate the first variable (x) from Rows 2 and 3:
     - Row 2: $R_2 = R_2 + 3 \cdot R_1$
     - Row 3: $R_3 = R_3 + 2 \cdot R_1$

$$\begin{bmatrix} 1 & 0.5 & -0.5 & 4 \\ 0 & 0.5 & 0.5 & 1 \\ 0 & 2 & 1 & 5 \end{bmatrix}$$

2. **Second Pivot (Row 2):**
   - Make the second pivot element 1 by dividing Row 2 by 0.5:

$$\begin{bmatrix} 1 & 0.5 & -0.5 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 2 & 1 & 5 \end{bmatrix}$$

   - Eliminate the second variable (y) from Row 3:
     - Row 3: $R_3 = R_3 - 2 \cdot R_2$

$$\begin{bmatrix} 1 & 0.5 & -0.5 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

3. **Third Pivot (Row 3):**
   - Make the third pivot element 1 by dividing Row 3 by -1:

$$\begin{bmatrix} 1 & 0.5 & -0.5 & 4 \\ 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$