Department of Computer and Software Engineering
SE: Machine Learning

Course Instructor: Dr. Ahmed Raza	**Date:** 12-February-2026
Day: Thursday	**Semester:** 6th

Lab 1: The Engine of Learning: Gradient Descent and KNN

Lab Title	CLO	BT	PLO	Weightage
The Engine of Learning: Gradient Descent and KNN				

Name	Roll Number	Report Marks/10	Viva Marks/5	Total Marks/15
Shehroz Muhammad Khan	BSSE23102			

Checked on: ____________________________

Signature: ____________________________

1.1 Learning Outcomes

After completing this Lab, the student will be able to:

- Implement gradient descent for scalar and vector-valued functions
- Understand convergence, divergence, and learning rate effects
- Implement K-Nearest Neighbors using distance geometry
- Compare loop-based vs vectorized implementations

1.2 Equipment

- PC with Python 3.x installed (Anaconda Distribution recommended).
- IDE of choice (Jupyter Notebook, Spyder, VS Code, or PyCharm).
- NumPy, Pandas, and Matplotlib libraries installed.

1.3 Theory & Background

Machine Learning heavily relies on optimization algorithms and geometric reasoning. In this lab, you will build machine learning algorithms (Gradient Descent and K-Nearest Neighbors) from scratch using math and NumPy. No sklearn. Only optimization, geometry, and engineering discipline.

1.3.1 Gradient Descent

Gradient Descent is an iterative optimization algorithm used to minimize a function by moving in the direction of steepest descent. The update rule is:

$$\theta_{n+1} = \theta_n - \alpha \nabla f(\theta_n)$$

where α is the learning rate and ∇f represents the gradient. The learning rate controls the step size and critically affects convergence behavior.

1.3.2 K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based learning algorithm. It classifies a data point based on the majority class among its k nearest neighbors in the feature space. The algorithm relies on distance metrics, typically Euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{N} \left(x_i - y_i\right)^2}$$

KNN has no explicit training phase; all computation happens at prediction time, making it memory-intensive for large datasets.

1.4 Submission Format

- Submit the provided filled in python script (.py), along with a manual in PDF format in a zip file.
- Naming convention: BSSE23XXX_SECTION#_ML_LABY (e.g., BSSE23001_A_ML_LAB1), where XXX is the remaining roll number and Y is the lab number.

PART A: Gradient Descent

Task A1: Scalar Gradient Descent [2 Marks]

Function: $f(x) = x^2 + 5$

1. Derive $f'(x)$ analytically.
2. Implement gradient descent starting from $x_0 = 10$.
3. Run for 20 iterations with learning rates: $\alpha = 0.1, 1.0, 1.1$
4. Identify which values converge or diverge and explain why.

Task A2: Gradient Descent on a 2D Surface [2 Marks]

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

1. Derive the gradient vector $\nabla f = [\partial f/\partial x, \partial f/\partial y]$.
2. Start from $(x_0, y_0) = (5, 5)$ with learning rate $\alpha = 0.1$.
3. Track parameter updates for at least 10 iterations.
4. Explain why convergence rate differs for x and y dimensions.

Task A3: Failure Mode [1 Mark]

1. Choose a learning rate that causes divergence.
2. Log the exploding values over iterations.
3. Explain divergence mathematically: why does the value grow instead of decrease?

Task A1: Scalar Gradient Descent

- **Derivation: $f'(x) = 2x$**
- **Convergence Behavior:**
  - **$\alpha = 0.1$ (Converges): The decay factor is $1 - 2(0.1) = 0.8$, so x shrinks toward 0.**
  - **$\alpha = 1.0$ (Oscillates): The factor is $1 - 2(1.0) = -1$, so x flips between 10 and -10.**
  - **$\alpha = 1.1$ (Diverges): The factor is $|1 - 2(1.1)| = |-1.2| > 1$, causing exponential growth.**

Task A2: Gradient Descent on a 2D Surface

- **Gradient: $\nabla f = [2x, 6y]$**
- **Convergence Rate: The y-dimension converges faster because its gradient coefficient (6) is larger than x's (2), creating a steeper slope that drives the value to zero more quickly.**

Task A3: Failure Mode

- **Mathematical Explanation: Divergence happens when the step size exceeds the function's stability limit. For y, the update is y_new = y(1 - 6α). If α = 0.4, the multiplier is |1 - 2.4| = 1.4. Since 1.4 > 1, the error grows by 40% every step.**

PART B: K-Nearest Neighbors (KNN)

Dataset: A synthetic 2D binary classification dataset generated using NumPy (please refer to function *generate_dataset* in the code file provided).
The dataset is linearly inseparable and reproducible via a fixed random seed.

Task B1: Euclidean Distance [1 Mark]

1. Implement Euclidean distance between two points using loops.
2. Implement Euclidean distance using NumPy vectorization (no loops).
3. Verify both implementations produce the same result.
4. Compare code clarity and efficiency.

Task B2: Distance Computation with Broadcasting [2 Marks]

1. Compute distances from a single test point to all training points.
2. No loops allowed, use NumPy broadcasting.
3. Explain the broadcasting expression in comments.

Task B3: Neighbor Selection (No Full Sorting) [1 Mark]

1. Use np.argpartition to find k nearest neighbors without full sorting.

2. Explain why full sorting with np.argsort is inefficient for large datasets.
3. Compare time complexity: O(n log n) vs O(n).

Task B4: KNN Prediction [2 Marks]

1. Retrieve the labels of the k nearest neighbors.
2. Perform majority voting to determine the predicted class.
3. Return the predicted class label.

Task B5 — Effect of k [1 Mark]

1. Test your KNN implementation with k = 1, k = 5, and k = n (total training points).
2. Identify which values lead to overfitting and which lead to under fitting.
3. Explain the behavior observed.

Task B1: Euclidean Distance

- **Comparison: The vectorized implementation is faster and cleaner because it uses optimized C-level routines (BLAS/LAPACK) within NumPy, avoiding the overhead of Python interpreter loops.**

Task B2: Broadcasting

- **Explanation: Broadcasting virtually replicates the smaller array (x_test) to match the shape of the larger array (X_train), allowing element-wise subtraction without physically copying data in memory.**

Task B3: Neighbor Selection

- **Efficiency: np.argpartition is O(n) because it only selects the top k elements, whereas np.argsort is O(n log n) because it sorts the entire array. Partitioning is significantly faster for large datasets.**

Task B5: Effect of k

- **k = 1 (Overfitting): High variance; the model captures noise and outliers, creating a jagged decision boundary.**
- **k = 5 (Balanced): Good trade-off; smooths out noise while preserving local structure.**

- **k = n (Underfitting): High bias; the model ignores local data entirely and always predicts the majority class of the dataset.**

PART C — Reflection [3 Marks]

1. Compare Gradient Descent vs KNN in terms of:
   - Training cost
   - Prediction cost
   - Memory usage
2. Which algorithm scales better to large datasets? Justify your answer.
3. Why is KNN rarely used in production ML systems despite its simplicity?

1. Comparison

- **Training Cost:** Gradient Descent is **High** (iterative optimization); KNN is **Zero** (lazy learning).
- **Prediction Cost:** Gradient Descent is **Low** (O(d)); KNN is **High** (O(nd)).
- **Memory Usage:** Gradient Descent is **Low** (stores weights); KNN is **High** (stores all data).

2. Scalability

- **Gradient Descent** scales better. Its prediction speed and model size are constant, whereas KNN becomes slower and heavier as the dataset size (n) increases.

3. Production Usage

- **KNN is rarely used** because it is too slow for real-time applications (latency increases with data) and requires excessive memory to store the training set.

Conclusion [1 Mark]

Write a brief summary of what you learned from this lab, focusing on the importance of understanding algorithmic foundations before using high-level libraries.

I learned that while "lazy" algorithms like KNN are simple to implement, "eager" algorithms like Gradient Descent are necessary for scalability. I also verified that vectorization is critical for performance in Python ML.

Assessment Rubrics

SE306L: Machine Learning Lab

Method: Lab reports and instructor observation during lab sessions

Outcome Assessed:

a. Ability to conduct experiments, as well as to analyze and interpret data (P).
b. Ability to use the techniques, skills, and modern engineering tools necessary for engineering practice (P).

Performance	Exceeds expectation (4-5)	Meets expectation (3-2)	Does not meet expectation (1)	Marks
1. Realization of Experiment [a, b]	Selects relevant equipment to the experiment, develops setup diagrams of equipment connections or wiring.	Needs guidance to select relevant equipment to the experiment and to develop equipment connection or wiring diagrams.	Incapable of selecting relevant equipment to conduct the experiment, equipment connection or wiring diagrams are	
2. Conducting Experiment [a, b]	Does proper calibration of equipment, carefully examines equipment moving parts, and ensures smooth operation and process.	Calibrates equipment, examines equipment moving parts, and operates the equipment with minor error.	Unable to calibrate appropriate equipment, and equipment operation is substantially wrong.	
3. Laboratory Safety Rules [a]	Respectfully and carefully observes safety rules and procedures	Observes safety rules and procedures with minor deviation.	Disregards safety rules and procedures.	
6. Data Collection [a]	Plans data collection to achieve experimental objectives, and conducts an orderly and a complete data collection.	Plans data collection to achieve experimental objectives, and collects complete data with minor error.	Does not know how to plan data collection to achieve experimental goals; data collected is incomplete and contain errors.	
7. Data Analysis [a]	Accurately conducts simple computations and statistical analysis using collected data;	Conducts simple computations and statistical analysis using collected data	Unable to conduct simple statistical analysis on collected data; no attempt	

	correlates experimental results to known theoretical values; accounts for measurement errors and parameters that affect experimental results.	with minor error; reasonably correlates experimental results to known theoretical values; attempts to account for measurement errors and parameters that affect experimental results.	to correlate experimental results with known theoretical values; incapable of explaining measurement errors or parameters that affect the experimental results.	
8. Computer Use [a]	Uses computer to collect and analyze data effectively.	Uses computer to collect and analyze data with minor error.	Does not know how to use computer to collect and analyze data.	
Total				

Faculty:

Name:__________________________

Signature:_______________________

Date: ___________________________