Week 4: Lab Programs

Question 1: Artificial Intelligence

Consider the **"TSP TW Dataset"** from the TSPLIB library, which provides coordinates of cities and distance matrices for various problems.
Dataset Link: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/

For simplicity, you can directly use the dataset `berlin52.tsp` for this lab. It contains the coordinates of 52 cities in Berlin.

Task:

- Test both Simple and Stochastic Hill Climbing on:
  - `berlin52.tsp` (52 cities).
  - `a280.tsp` (280 cities) from TSPLIB.
- Tabulate the results for each dataset, including:
  - Total distance.
  - Execution time.
  - Number of iterations to converge.
- Identify which algorithm performs better for each dataset size.
- Plot the total distance vs. iterations for both algorithms on the same graph.
- Comment on the convergence patterns for Simple and Stochastic Hill

Deliverable:

Performance table for berlin52.tsp and a280.tsp.

Q2. Deep Learning Lab Exercise: Building a CNN for MNIST

Objective:
Implement a Convolutional Neural Network (CNN) to classify handwritten digits in the MNIST dataset.

Task:

- Modify the existing ANN code to build a CNN with the following structure:
  - Input layer.
  - Two convolutional layers (32 filters of size 3x3 and 64 filters of size 3x3).
  - MaxPooling layer after each convolutional layer.
  - Dropout layer (with a dropout rate of 0.25).
  - Dense layers for classification.
- Train the CNN for 10 epochs and report the accuracy.
- Train both ANN and CNN on the MNIST dataset for 10 epochs each.
- Compare:
  - Training time for each epoch.
  - Accuracy on the training and testing datasets.

- Plot the loss and accuracy curves for both models.

Deliverables:

- Python code for the CNN architecture.
- Final training and testing accuracies.
- A performance comparison table.
- Plots of loss vs. epochs and accuracy vs. epochs for both models.
- Insights on how and why CNN improves performance over ANN.