Deep Learning (DL) and High Performance Computing (HPC) - Questions with Answers

1. What is Linear Regression

Answer: Linear Regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data.

2. Example of Linear Regression

Answer: Example: Predicting house prices based on features like area and number of rooms using the equation Y = mX + b.

3. Concept of Deep Neural Network

Answer: A Deep Neural Network (DNN) is an artificial neural network with multiple layers between the input and output layers, allowing the model to learn complex patterns.

4. How Deep Neural Network Work

Answer: DNNs work by passing inputs through multiple layers of neurons, each performing linear transformations followed by non-linear activation functions.

5. Code Explanation with Output (Regression)

Answer: Using TensorFlow or PyTorch, you can build and train a regression model, then evaluate its performance using metrics like MSE.

6. What is Classification

Answer: Classification is the process of predicting the category or class of given data points.

7. Example of Classification

Answer: Example: Email spam detection where emails are classified as 'spam' or 'not spam'.

8. How Deep Neural Network Work on Classification

Answer: DNNs perform classification by outputting probabilities for each class using a softmax function in the final layer.

9. Code Explanation with Output (Classification)

Answer: Train a classification model using cross-entropy loss and evaluate it using accuracy and confusion matrix.

10. What is DFS?

Answer: Depth-First Search (DFS) is a graph traversal algorithm that explores as far as possible along each branch before backtracking.

11. Example of DFS

Answer: Example: Starting from node A, DFS visits deeper nodes before backtracking to explore siblings.

12. Concept of OpenMP

Answer: OpenMP is an API for shared memory multiprocessing programming in C, C++, and Fortran, enabling parallelism using compiler directives.

13. How Parallel DFS Work

Answer: Parallel DFS uses threads to explore different branches concurrently, but care must be taken due to its recursive nature.

14. What is BFS?

Answer: Breadth-First Search (BFS) is a graph traversal algorithm that explores all neighbor nodes at the present depth before moving to the next level.

15. Example of BFS

Answer: Example: Starting from node A, BFS visits all immediate neighbors before moving deeper.

16. How Parallel BFS Work

Answer: Parallel BFS distributes frontier nodes across threads and processes their neighbors concurrently.

17. Code Explanation with Output (BFS)

Answer: Parallel BFS can be implemented using queues with OpenMP. Output includes traversal order and time comparisons.

18. What is Bubble Sort? Use of Bubble Sort

Answer: Bubble Sort is a simple sorting algorithm that repeatedly steps through the list, compares adjacent elements and swaps them if in the wrong order.

19. Example of Bubble Sort?

Answer: Example: Sorting [5, 1, 4] results in [1, 4, 5] after multiple iterations.

20. How Parallel Bubble Sort Work

Answer: Parallel Bubble Sort divides the list into segments and sorts them concurrently using OpenMP.

21. How to measure the performance of sequential and parallel algorithms?

Answer: Measure execution time and compute speedup = Sequential Time / Parallel Time. Analyze efficiency and scalability.

22. What is Merge? Use of Merge Sort

Answer: Merge Sort is a divide-and-conquer algorithm that splits the array, sorts recursively, and merges the results.

23. Example of Merge Sort?

Answer: Example: Sorting [3, 1, 2] involves dividing to [3], [1, 2], sorting to [1, 2], then merging to [1, 2, 3].

24. How Parallel Merge Sort Work

Answer: Parallel Merge Sort uses threads to concurrently sort sub-arrays and merge them using OpenMP.

25. What is parallel reduction and its usefulness for mathematical operations on large data?

Answer: Parallel reduction combines elements using operations like sum, max in a parallelized way, boosting performance for large data.

26. How do parallel reduction algorithms for Min, Max, Sum, and Average work, and what are their advantages and limitations?

Answer: They divide data among threads, compute partial results, and combine them. They offer speedup but may face load balancing and synchronization issues.