Name:

Enrolment No:

UPES

End Semester Examination, Dec 2024

Course: Soft Computing | **Semester: III**

Program: MCA | **Time: 03 hrs.**

Course Code: CSAI8009P | **Max. Marks: 100**

Instructions: *Please attempt according to the provided time and given weightage.*
Start Answering on the new page and mention the question number clearly on the left margin.

SECTION A
(5Qx4M=20Marks)

S. No.		Marks	CO
Q 1	Explain Soft Computing.	**4**	**CO1**
Q 2	Explain different types of distances and cost functions used in ML.	**4**	**CO5**
Q 3	Explain the use of activation function in Neural networks with examples.	**4**	**CO2**
Q 4	Explain Fuzzy sets and Crisp sets. Also, explain different types of membership functions.	**4**	**CO2**
Q 5	Explain the Chain Rule and Elitism.	**4**	**CO5**

SECTION B
(4Qx10M= 40 Marks)

Q 6	Design an AND gate using Perceptron, start with random weights, and use the weight update algorithm to adjust weights. **OR** Compare and contrast between biological neurons and artificial neurons. Show the basic mathematical model of perceptron.	**10**	**CO3, CO2**
Q 7	Derive an expression for gradient descent with backpropagation (BP)	**10**	**CO2**
Q 8	Obtain the truth table for given expressions and determine whether they are tautology or contradiction. i) $(P \vee Q) \Longrightarrow (\sim P)$ ii) $(P \Longrightarrow Q) = (\neg P \vee Q)$ iii) $((P \Longrightarrow Q) \wedge (Q \Longrightarrow P) = (P = Q))$	**10**	**CO1**

Q 9	Explain defuzzification and commonly used defuzzification methods.	**10**	**CO2**
Q 1**SECTION-C** **(2Qx20M=40 Marks)**			
Q 10	Explain the gradient descent algorithm and derive an expression to update weights and biases. **OR** Design a neural network to classify 60,000 handwritten digits, analyze its computational complexity, and justify the choice of architecture.	**20**	**CO5, CO3**
Q 11	Define genetic algorithm. Provide a Side-by-side comparison of biological genetics and genetic algorithms. **OR** Explain with an example where the genetic algorithm is useful. Write Pseudo (or Python) code to implement a Genetic algorithm to solve the knapsack problem (or any other example problem).	**20**	**CO1, CO3**