1. What is soft computing? How it differs from traditional hard computing? [3+2=5]

Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve actability, robustness, and low solution cost. Its principal constituents are fuzzy logic, neurocomputing, and probabilistic reasoning. Soft computing is likely to play an increasingly important role in many application areas, including software engineering. The role model for Soft computing is the human mind. The guiding principle for soft computing is:

- Exploit the tolerance for imprecision, uncertainty, partial truth, and approximation to achieve tractability, robustness and low solution cost.
- Learning from experimental data

Approximation: here the model features are similar to the real ones, but not the same.
Uncertainty: here we are not sure that the features of the model are the same as that of the entity (belief).
Imprecision: here the model features (quantities) are not the same as that of the real ones, but close to them.

Soft Computing	Hard Computing
Soft Computing is liberal of inexactness, uncertainty, partial truth and approximation.	Hard computing needs an exactly state analytic model.
Soft computing produces approximate results.	Hard computing produces precise results.
Soft computing will emerge its own programs.	Hard computing requires programs to be written.
Soft computing works on ambiguous and noisy data.	Hard computing works on exact data.
Soft computing is stochastic in nature.	Hard computing is deterministic in nature.
Soft computing will use multivalued logic.	Hard computing uses two-valued logic.

2. What are the differences between supervised and unsupervised learning algorithms? Explain briefly. [5]

Supervised learning: Supervised learning is the learning of the model where with input variable (say, x) and an output variable (say, Y) and an algorithm to map the input to the output.

That is, $Y = f(X)$

The basic aim is to approximate the mapping function (mentioned above) so well that when there is a new input data (x) then the corresponding output variable can be predicted.

It is called supervised learning because the process of a learning (from the training dataset) can be thought of as a teacher who is supervising the entire learning process. Thus, the "learning algorithm" iteratively makes predictions on the training data and is corrected by the "teacher", and the learning stops when the algorithm achieves an acceptable level of performance (or the desired accuracy).

Unsupervised Learning: Unsupervised learning is where only the input data (say, X) is present and no corresponding output variable is there.

The main aim of Unsupervised learning is to model the distribution in the data in order to learn more about the data.

It is called so, because there is no correct answer and there is no such teacher (unlike supervised learning). Algorithms are left to their own devises to discover and present the interesting structure in the data.

Supervised Learning	Unsupervised Learning
Uses Known and Labeled Data as input	Uses Unknown Data as input

Very Complex	Less Computational Complexity
Uses off-line analysis	Uses Real Time Analysis of Data
Number of Classes are known	Number of Classes are not known
Accurate and Reliable Results	Moderate Accurate and Reliable Results

3. Illustrate Back-Propagation learning algorithm by means of a flowchart. [5]

4. What are the differences between Perceptron and Gradient-descent search? What do you mean by Hebb learning? [3+2=5]

5. What do you mean by elitism selection properties of GA? What are the pros and cons of it? [3+2=5]

Elitist selection is a selection strategy where a limited number of individuals with the best fitness values are chosen to pass to the next generation, avoiding the crossover and mutation operators.
Pros: Elitism prevents the random destruction by crossover or mutation operators of individuals with good genetics.
Cons: The number of elite individuals should not be too high, otherwise the population will tend to degenerate.

6. What is the role of activation function in learning? [5]

The purpose of an activation function is to add some kind of non-linear property to the function, which is a neural network. Without the activation functions, the neural network could perform only linear mappings from inputs x to the outputs y.

Without the activation functions, the only mathematical operation during the forward propagation would be dot-products between an input vector and a weight matrix. Since a single dot product is a linear operation, successive dot products would be nothing more than multiple linear operations repeated one after the other. And successive linear operations can be considered as a one single learn operation.

In order to be able to compute really interesting stuff, neural networks must be able to approximate nonlinear relations from input features to output labels. Usually, the more complex the data is we are trying to learn something from, the more non-linear the mapping of features to the ground truth label is.

A neural network without any activation function would not be able to realize such complex mappings mathematically and would not be able to solve tasks we want the network to solve.

7. Discuss pros and cons of different defuzzification methods with a proper example. [5]

8. Consider three fuzzy sets given by: 5

P = {(x , 1),(y, 0.2),(z, 0.5)}
Q = {(a, 0.9), (b, 0.4), (c, 0.9)}
R = {(x, 0.1),(y, 0.2),(z, 0.7)}
Find the fuzzy relation for the Cartesian Product of P and Q i.e. R = P X Q.

R = {((x,a),0.9), ((x,b),0.4), ((x,c),0.9), ((y,a),0.2), ((y,b),0.2), ((y,c),0.2), ((z,a),0.5), ((z,b),0.4), ((z,c),0.5)}