NAME : ANURAN CHAKRABORTY
ROLL NO.: 20 REG. NO.: 135926

SUBJECT : SOFT COMPUTING CT-1.
DATE : 16/05/20.

Q1. What is soft computing? How it differs from traditional hard computing? [3+2]

Soft computing is a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, 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 : the model features are similar to the real ones, but not the same.
- Uncertainty : we are not sure that the features of the model are same as that of the entity.
- Imprecision : the model features are not the same as the real ones, but close to them.

SC = Evolutionary Computing + Neural Network + Fuzzy Logic

Soft Computing	Hard Computing
Soft computing is liberal of inexactness, uncertainty, partial truth, and approximation.	Hard computing needs an exact state analytic model.
Produces approximate results.	Produces precise results
Will use multivalued logic.	Uses two-valued logic.
Stochastic in nature	Deterministic in nature.
Works on ambiguous and noisy data.	Works on exact data.

Q2. What are the differences between supervised and unsupervised learing algorithms? Explain. [5]

Supervised Learning	Unsupervised Learning
It is the learning of the model where there is an input variable (X) and an output variable (Y). The supervised learning tries to learn the mapping $Y = f(X)$.	In unsupervised learning there is only the input data (X) and no corresponding output variable is there. The algorithm tries to infer information from only the input data.
Basic aim is to approximate the mapping function so well that when there is a new input data (X) then the corresponding output can be predicted.	Main aim is to model the distribution in the data in order to learn more about the data.

Supervised Learning	Unsupervised learning
• It is called so because the process of a learning can be thought of as a teacher who is supervising the entire learning process by correcting it.	• It is called so because there is no correct answer and there is no teacher to correct it. Algorithms discover the interesting structure in the data on their own.
• Number of classes are known and also each class labels are known.	• Either number of classes are not known, or the class labels are unknown or both.
• Examples: SVM, Neural Networks, etc.	• Examples: Clustering algorithms such as K-means clustering, etc.

Q6. 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 operation during forward propagation would be dot-products between an input vector and a weight matrix. A single dot product is a linear operation. So, successive dot products is nothing but multiple linear operations which is ~~nothing~~ effectively a single linear operation.

But in order to be able to compute complex functions and learn a better mapping neural networks must be able to approximate non-linear relations from input features to output labels. Usually, the more complex the data we are trying to learn something from, the more non-linear the mappings of features to the ground truth label is. Without non-linearity the neural network would fail to learn such complex mappings.

Q5. What do you mean by the elitism selection properties of GA? What are the pros and cons of it? [3+2]

Genetic Algorithms are evolutionary algorithms where some initial trial solutions are generated called ~~popup~~ population. Over the course of the next generations these solutions are changed by crossover and mutation giving rise to a ~~ee~~ new population in each generation. Now, some of these members in each population are selected on the basis of some fitness criteria, ~~the~~ for ~~the~~ its contribution to the next generation. Elitism selection involves copying a small proportion of the fittest ~~can~~ members to the next generation avoiding crossover and mutation. Candidate solutions that are preserved through elitism remain eligible for selection as parents when breeding the remainder of the next generation.

Pros :

→ Elitism can have a dramatic effect on the performance because it ensures that the algorithm does not spend time discovering previous fit solutions. Thus convergence speed increases.

→ The solution quality is not degraded from one generation to the next as it prevents random destruction of best candidates

Cons :

→ If the percentage of elitism is high there will be less diversity in population. Thus, the solution will tend to degenerate.

→ Elitism increases convergence speed, so it can lead to premature convergence which can result in the algorithm getting stuck in the local minima (or maxima) instead of the global minima (or maxima).

Q8. Consider the fuzzy sets : [5]

$P = \{(x,1), (y,0.2), (z,0.5)\}$

$Q = \{(a,0.9), (b,0.4), (c,0.9)\}$.

Find the fuzzy relation for the Cartesian Product of P and Q i.e. $R = P \times Q$.

Suppose $\mu_A(x)$ denote the membership value of x in set A. Then for cartesian product

$$\mu_{A\times B}(x,y) = \min(\mu_A(x), \mu_B(y))$$

for all $x \in A$ and $y \in B$

$$\therefore P \times Q = \{[(x,a),0.9], [(x,b),0.4], [(x,c),0.9],$$
$$[(y,a),0.2], [(y,b),0.2], [(y,c),0.2],$$
$$[(z,a),0.5], [(z,b),0.4], [(z,c),0.5]\}$$