END TERM EXAMINATION [MAY-JUNE 2018]
EIGHTH SEMESTER [B.TECH]
SOFT COMPUTING [ETIT-410]

Time: 3 Hrs. **Max. Marks: 75**

Note: Attempt any five questions including Q. no. 1 which is compulsory.

Q.1. Attempt following in brief:

Q.1. (a) Differentiate between hard and soft computing. **(5)**

Ans.

S.No.	Soft Computing	Hard Computing
1.	Soft Computing is liberal of inexactness, uncertainty, partial truth and approximation.	Hard computing needs a exactly state analytic model.
2.	Soft Computing relies on formal logic and probabilistic reasoning.	Hard computing relies on binary logic and crisp system.
3.	Soft computing has the features of approximation and dispositionality.	Hard computing has the features of exactitude(precision) and categoricity.
4.	Soft computing is stochastic in nature.	Hard computing is deterministic in nature.
5.	Soft computing works on ambiguous and noisy data.	Hard computing works on exact data.
6.	Soft computing can perform parallel computations.	Hard computing performs sequential computations.
7.	Soft computing produces approximate results.	Hard computing produces precise results.
8.	Soft computing will emerge its own programs.	Hard computing requires programs to be written.
9.	Soft computing incorporates randomness .	Hard computing is settled.
10.	Soft computing will use multivalued logic.	Hard computing uses two-valued logic.

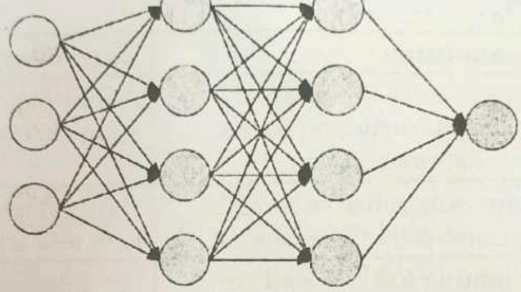
Q.1. (b) Draw an architecture of Neural Network and explain. **(5)**

Ans. Neural Network Architecture: Neural Networks are complex structures made of artificial neurons that can take in multiple inputs to produce a single output. This is the primary job of a Neural Network – to transform input into a meaningful output. Usually, a Neural Network consists of an input and output layer with one or multiple hidden layers within. It is also known as Artificial Neural Network or ANN. ANN architecture in Neural Network functions just like a human brain and is very important.

In a Neural Network, all the neurons influence each other, and hence, they are all connected. The network can acknowledge and observe every aspect of the dataset at hand and how the different parts of data may or may not relate to each other. This is how Neural Networks are capable of finding extremely complex patterns in vast volumes of data. In a Neural Network, the flow of information occurs in two ways –

•**Feedforward Networks:** In this model, the signals only travel in one direction, towards the output layer. Feedforward Networks have an input layer and a single output layer with zero or multiple hidden layers. They are widely used in pattern recognition.

•**Feedback Networks:** In this model, the recurrent or interactive networks use their internal state (memory) to process the sequence of inputs. In them, signals can travel in both directions through the loops (hidden layer/s) in the network. They are typically used in time-series and sequential tasks.



Input layer hidden layer 1 hidden layer 2 output layer

Q.1. (c) What is Hopfield network? Explain the algorithm to store and recall a set of bipolar patterns in Hopfield network. **(5)**

Ans. The Hopfield Neural Networks, invented by Dr John J. Hopfield consists of one layer of 'n' fully connected recurrent neurons. It is generally used in performing auto association and optimization tasks. It is calculated using a converging interactive process and it generates a different response than our normal neural nets.

Following are some important points to keep in mind about discrete Hopfield network –

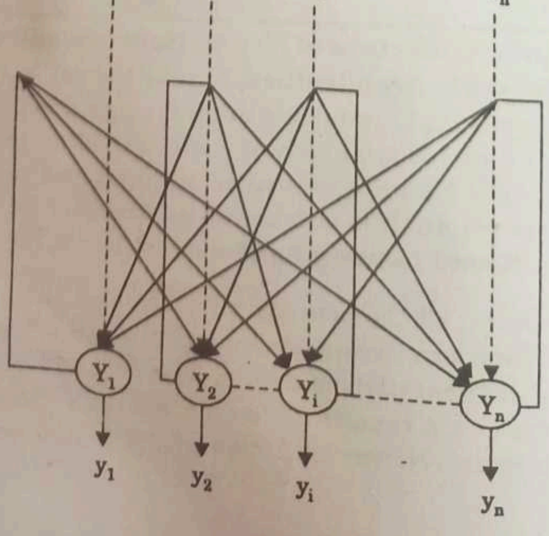
•This model consists of neurons with one inverting and one non-inverting output.

•The output of each neuron should be the input of other neurons but not the input of self.

•Weight/connection strength is represented by w_{ij}.

•Connections can be excitatory as well as inhibitory. It would be excitatory, if the output of the neuron is same as the input, otherwise inhibitory.

•Weights should be symmetrical, i.e. $w_{ij} = w_{ji}$



The output from Y_1 going to Y_2, Y_i and Y_n have the weights w_{12}, w_{1i} and w_{1n} respectively. Similarly, other arcs have the weights on them.

Training Algorithm

During training of discrete Hopfield network, weights will be updated. As we know that we can have the binary input vectors as well as bipolar input vectors. Hence, in both the cases, weight updates can be done with the following relation

Case 1 - Binary input patterns

For a set of binary patterns s p , p = 1 to P

Here, s p $= s_1\, p, s_2\, p ,..., s_i\, p ,..., s_n\, p$

Weight Matrix is given by

$$w_{ij} = \sum_{p=1}^{P}\left[2s_i(p)-1[2s_j(p)-1]\right] \text{ for } i \neq j$$

Case 2 - Bipolar input patterns

For a set of binary patterns s p, p = 1 to P

Here, s p $= s_1\, p, s_2\, p,..., s_i\, p ,..., s_n\, p$

Weight Matrix is given by

$$w_{ij} = \sum_{p=1}^{P}\left[s_i(p)[s_j(p)\right] \text{ for } i \neq j$$

Testing Algorithm

Step 1 – Initialize the weights, which are obtained from training algorithm by using Hebbian principle.

Step 2 – Perform steps 3-9, if the activations of the network is not consolidated.

Step 3 – For each input vector X, perform steps 4-8.

Step 4 – Make initial activation of the network equal to the external input vector X as follows –

$y_i = x_i$ for i = 1 to n

Step 5 - For each unit Y_i, perform steps 6-9.

Step 6 - Calculate the net input of the network as follows-

$$y_i n_i = x_i + \sum_j y_j w_{ji}$$

Step 7 - Apply the activation as follows over the net input to calculate the output - Here θ_i is the threshold.

Step 8 - Broadcast this output y_i to all other units.

Step 9 - Test the network for conjunction.

Q.1. (d) Explain the error correction process and gradient descent rule. (5)

Ans. Error-Correction Learning, used with supervised learning, is the technique of comparing the system output to the desired output value, and using that error to direct the training. In the most direct route, the error values can be used to directly adjust the tap weights, using an algorithm such as the back propagation algorithm. If the system output is y, and the desired system output is known to be d, the error signal can be defined as:

$$e = d - y$$

Error correction learning algorithms attempt to minimize this error signal at each training iteration.

The gradient descent algorithm is not specifically an ANN learning algorithm. It has a large variety of uses in various fields of science, engineering, and mathematics. However, we need to discuss the gradient descent algorithm in order to fully understand the back propagation algorithm. The gradient descent algorithm is used to minimize an error function g(y), through the manipulation of a weight vector w. The cost function should be a linear combination of the weight vector and an input vector x. The algorithm is:

$$w_{ij}[n+1] = w_{ij}[n] + \eta g(w_{ij}[n])$$

Here, η is known as the step-size parameter, and affects the rate of convergence of the algorithm. If the step size is too small, the algorithm will take a long time to converge. If the step size is too large the algorithm might oscillate or diverge.

The gradient descent algorithm works by taking the gradient of the weight space to find the path of steepest descent. By following the path of steepest descent at each iteration, we will either find a minimum, or the algorithm could diverge if the weight space is infinitely decreasing. When a minimum is found, there is no guarantee that it is a global minimum, however.

Q.1. (e) Find $A \cup B$ and complement of $A \cup B$ for the following two fuzzy sets:

$$A = \{1/1.0 + 0.75/1.5 + 0.3/2.0 + 0.15/2.5 + 0/3\}$$
$$B = \{1/1.0 + 0.6/1.5 + 0.2/2.0 + 0.1/2.5 + 0/3\}$$

Ans. $\underset{\sim}{A} \cup \underset{\sim}{B} = \max\{M_{\underset{\sim}{A}}(x), M_{\underset{\sim}{B}}(x)\}$

$A \cup B = \max(\mu_A(x), \mu_B(x))$

$$= \max\left\{\left(\frac{1}{1.0}+\frac{0.75}{1.5}+\frac{0.3}{2.0}+\frac{0.15}{2.5}+\frac{0}{3}\right)\left(\frac{1}{1.0}+\frac{0.6}{1.5}+\frac{0.2}{2.0}+\frac{0.1}{2.5}+\frac{0}{3}\right)\right\}$$

$$\underset{\sim}{A} \cup \underset{\sim}{B} = \left\{\frac{1}{1.0}+\frac{0.75}{1.5}+\frac{0.3}{2.0}+\frac{0.15}{2.5}+\frac{0}{3}\right\}$$

$$\overline{\underset{\sim}{A} \cup \underset{\sim}{B}} = \overline{\underset{\sim}{A}} \cap \overline{\underset{\sim}{B}} = \min\{M_{\underset{\sim}{A}}(x), M_{\underset{\sim}{B}}(x)\}$$

$$= \min\left\{\left(\frac{1}{1.0}+\frac{0.75}{1.5}+\frac{0.3}{2.0}+\frac{0.15}{2.5}+\frac{0}{3}\right)\left(\frac{1}{1.0}+\frac{0.6}{1.5}+\frac{0.2}{2.0}+\frac{0.1}{2.5}+\frac{0}{3}\right)\right\}$$

$$= \left\{\frac{1}{1.0}+\frac{0.6}{1.5}+\frac{0.2}{2.0}+\frac{0.1}{2.5}+\frac{0}{3}\right\}$$

Q.2. (a) Differentiate between supervised and unsupervised learning. Give one example of each. (6

Ans. Refer Q. No. 3 First Term Exam 2017.

Q.2. (b) Describe McCulloch-Pitts Neuron. Implement "AND" function using McCulloch-Pitts Neuron. (6.5

Ans. The idea of the simple neuron model first emerged in the 1940s with the work of McCulloch and Pitts, The cybernetics movement, attempted to combine biology, psychology, engineering and mathematics suiting in architectures for networks of neurons which would perform a number of cast in 1949, Hebb put forward the theory of neural networks developing internal representation related to experience

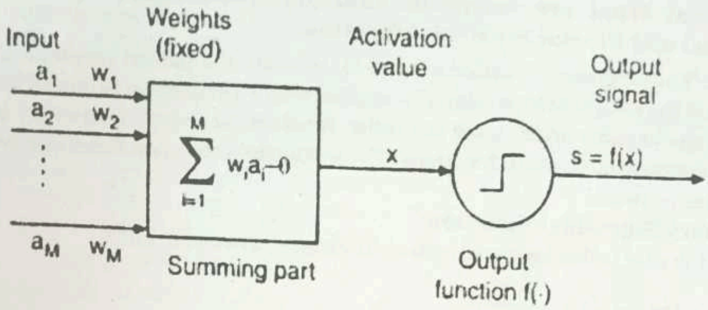


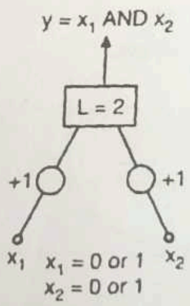
Fig. McCulloch and Pits Model of Neuron

In the 1950s, research continued initially into the development of networks to perform specific tasks but this changed and the goal became to develop machines that could learn. By the end of that decade there here had been a lack of significant developments and work in this field diminished considerably.

The perceptron effectively splits the input patterns into two distinct regions with one region being represented by a 1 on the output and the other a 0. Rosenblatt's training algorithm for the perceptron would converge if the input patterns to the perceptron were linearly separable. The perceptron would therefore approximate the decision boundary between the two classes of outputs.

Perceptrons were successfully trained to perform certain tasks but there were failures that could not be overcome, Minsky and Papert pointed out the serious shortcomings of perceptrons and interest in the study of neural networks again declined,

The 'AND' function using McCulloch-Pits neuron is shown below



AND gate

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Q.3. (a) What are activation function? Differentiate between Binary Sigmodial and Bipolar Sigmoidal function. (6)

Ans. The activation function is used to calculate the output response of a neuron. The sum of the weighted input signal is applied with an activation to obtain the response. For neurons in same layer, same activation functions are used. There may be linear as well as nonlinear activation functions. The nonlinear activation functions are used in a multilayer network.

Binary Sigmoidal Function

This is also called logistic function. It ranges between 0 and 1.

$$f(x) = \log \text{sig}\,(x) = \frac{1}{1+\exp^{-x}}$$

If f(x) is differentiated we get,

$$f(x) = f(x)\,[1 - f(x)]$$

Fig. Shows the binary sigmoidal function.

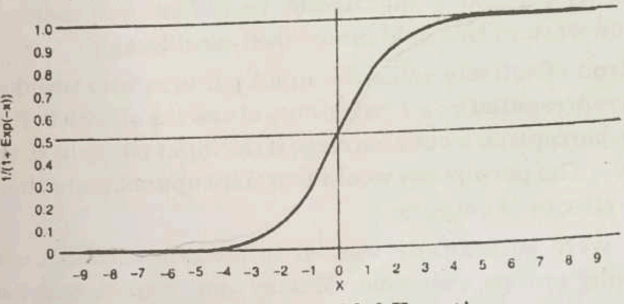


Fig. Binary Sigmoidal Functions

The desired range here is between +1 and –1. The function is related to the hyperbolic tangent function. The bipolar sigmoidal function is given as,

$$y(x) = 2f(x) - 1$$

Substituting the value of f(x) we get,

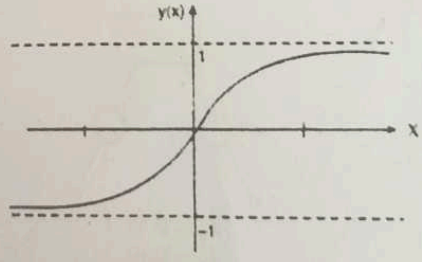
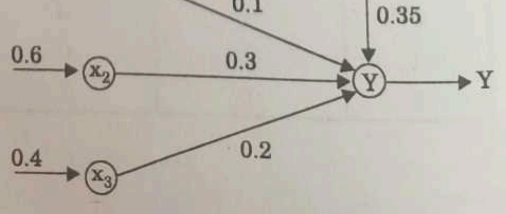


Fig. Bipolar Sigmoidal Function

Q.3. (b) Obtain the output of the neuron Y for the network shown in following figure using Binary Sigmodial and Bipolar Sigmoidal function. (6)



Ans. The given network has three input neurons with bias and one output neuron, These form a layer network,

The inputs are given as,

$[x_1, x_2, x_3] = [0.8, 0.6, 0.4]$

The weights are,

$[w_1, w_2, w_3] = [0.1, 0.3, -0.2]$

The net input can be calculated as,

$y_{in} = b + \sum_{i=1}^{n} (x_i w_i)$

$y_{in} = 0.35 + 0.8 \times 0.1 + 0.6 \times 0.3 + 0.4 \times (-0.2)$

$y_{in} = 0.35 + 0.08 + 0.18 - 0.08 = 0.53$

(i) For Binary Sigmoidal Function,

$$y = f(y_{in}) = \frac{1}{1+e^{-y_{in}}} = \frac{1}{1+e^{-0.53}} = 0.62$$

(ii) For Bipolar Sigmoidal activation function,

$$y = f(y_{in}) = \frac{2}{1+e^{-y_{in}}} - 1 = \frac{1-e^{-y_{in}}}{1+e^{-y_{in}}}$$

$$y = \frac{1-e^{-0.53}}{1+e^{-0.53}} = 0.259$$

Q.4. (a) What are Fuzzy Set? Enlist and explain various operators on Fuzzy Set. What do you mean by Lambda-Cut? (6)

Ans. Fuzzy sets support a flexible sense of membership of elements to a set. A fuzzy set is defined as follows:

If X is universe of discourse and x is a particular element of X, then a fuzzy set. A defined on X may be written as a collection of ordered pairs,

$$A = \{(x, \mu_A(x)), x\hat{I}X\}$$

where each pair $(x, \mu_A(x)$ is called a singleton.

Operations on Fuzzy set are:

(i) $\mu_{\tilde{A}\cup\tilde{B}}(x) = \max(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$

(ii) $\mu_{\tilde{A}\cap\tilde{B}}(x_1) = \min(\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x))$

(iii) $\mu_{\bar{\tilde{A}}}(x) = 1 - \mu_{\tilde{A}\cap\tilde{B}}(x)$

(iv) $\mu_{\tilde{A}\cdot\tilde{B}}(x) = \mu_{\tilde{A}}(x)\mu_{\tilde{B}}(x)$

(v) $\mu_{a\cdot\tilde{A}}(x) = a.\mu_{\tilde{A}}(x)$

(vi) $\mu_{A\alpha}(x) = (\mu_{\tilde{A}}(x))^{\alpha}$

(vii) $\tilde{A} - \tilde{B} = (\tilde{A} \cap \tilde{B}^c)$

(viii) $\tilde{A} \oplus \tilde{B} = (\tilde{A} \cap \tilde{B}) \cup (\tilde{A} \cap \tilde{B}^c)$

Q.4. (b) What is fuzzy relation? Draw a bipartite and simple fuzzy graph of the following relation X = {X1, X2, X3, X4} **(6.5)**

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{array}\begin{array}{c} \begin{array}{cccc} x1 & x2 & x3 & x4 \end{array} \\ \begin{pmatrix} 0.2 & 0 & 0.5 & 0 \\ 0 & 0.3 & 0.7 & 0.8 \\ 0.1 & 0 & 0.4 & 0 \\ 0 & 0.6 & 0 & 1 \end{pmatrix} \end{array}$$

Ans. Fuzzy relations elements of one universe (say X) to those of another universe (say Y) through the Cartesian product of the two universes. These can also be referred to as fuzzy sets defined on universal sets, which are Cartesian products.

A fuzzy relation is based on the concept that everything is related to some extent or unrelated.

A fuzzy relation is a fuzzy set defined on the Cartesian product of classical $\{X_1, X_2, \ldots X_n\}$ where tuples $(x_1 x_2, \ldots, x_n)$ may have varying degrees of $\mu_R(x_1 x_2, \ldots, x_n)$ within the relation, That is,

$$R(X_1, X_2, .., X_n) = \int_{X_1 \times X_2 x \ldots x X_n} \mu_R(x_1, x_2, \ldots, xn) \mid (x_1, x_2, \ldots, x_n), \quad x_i \in X_i$$

A fuzzy relation between two sets X and Y is called binary fuzzy relation and is denoted by R(X, Y). A binary relation R(X, Y) is referred to as bipartite graph when $X \neq Y$. The binary relation on a single set X is called directed graph or digraph. This relation occurs when X=Y and is denoted as R(X, X) or $R(X^2)$.

The bipartite graph and simple fuzzy graph of B(X, X) is shown in Figures below:

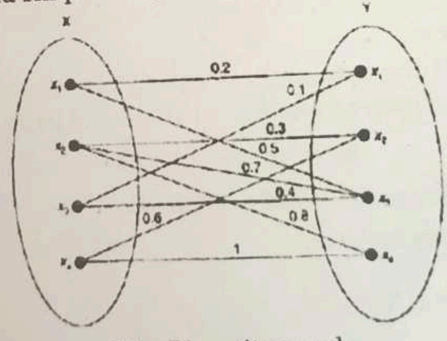


Fig. Bipartite graph

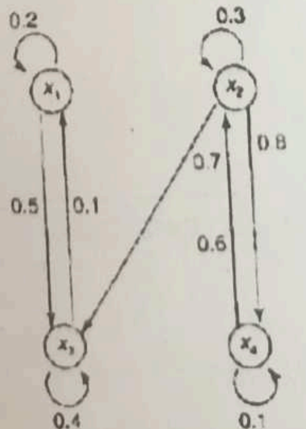


Fig. Simple fuzzy graph

Q.5. (a) What is defuzzification method? Enlist and explain various defuzzification methods. **(6)**

Ans. Refer Q.no. 5 (a) of End Term Exam 2018.

Q.5. (b) What is fuzzy inference system? Explain all types of is fuzzy inference system. What is fuzzy preposition? **(6.5)**

Ans. Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned.

Following are the two important methods of FIS, having different consequent of fuzzy rules –

- Mamdani Fuzzy Inference System
- Takagi-Sugeno Fuzzy Model (TS Method)

Mamdani Fuzzy Inference System

This system was proposed in 1975 by Ebhasim Mamdani. Basically, it was anticipated to control a steam engine and boiler combination by synthesizing a set of fuzzy rules obtained from people working on the system.

Steps for Computing the Output

Following steps need to be followed to compute the output from this FIS –

Step 1 – Set of fuzzy rules need to be determined in this step.

Step 2 – In this step, by using input membership function, the input would be made fuzzy.

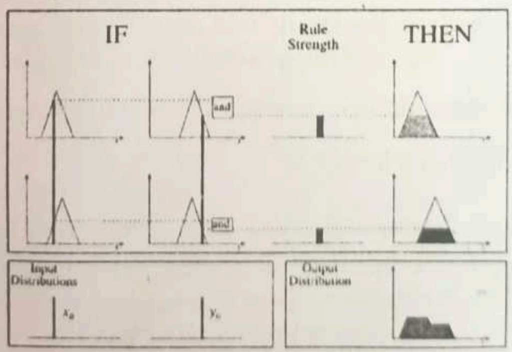
Step 3 – Now establish the rule strength by combining the fuzzified inputs according to fuzzy rules.

Step 4 – In this step, determine the consequent of rule by combining the rule strength and the output membership function.

Step 5 – For getting output distribution combine all the consequents.

Step 6 – Finally, a defuzzified output distribution is obtained.

Following is a block diagram of Mamdani Fuzzy Interface System.



Takagi-Sugeno Fuzzy Model (TS Method)

This model was proposed by Takagi, Sugeno and Kang in 1985. Format of this rule is given as –

$$\text{IF } x \text{ is A and } y \text{ is B THEN } Z = f(x, y)$$

Here, AB are fuzzy sets in antecedents and $z = f(x, y)$ is a crisp function in the consequent.

Fuzzy Inference Process

The fuzzy inference process under Takagi-Sugeno Fuzzy Model (TS Method) works in the following way.

Step 1: Fuzzifying the inputs – Here, the inputs of the system are made fuzzy.

Step 2: Applying the fuzzy operator – In this step, the fuzzy operators must be applied to get the output.

Q.6. (a) What are genetic algorithms? How Mutation, Selection and Crossover works in genetic algorithms? Explain. (6)

Ans. Refer Q.no. 1 (f) of End Term Exam 2018.

Q.6. (b) What are linguistic variables? How they are different from numeric variable. (6.5)

Ans. Refer Q.no. 3 (a) of First Term Exam 2018.

Q.7. (a) What is learning in neural networks? Explain linear separable and non-linearly separable pattern with example. (6)

Ans. Learning, in artificial neural network, is the method of modifying the weights of connections between the neurons of a specified network. Learning in ANN can be classified into three categories namely supervised learning, unsupervised learning, and reinforcement learning.

We say they're separable if there's a classifier whose decision boundary separates the positive objects from the negative ones. If such a decision boundary is a linear function of the features, we say that the classes are linearly separable.

For example, consider a dataset with two features x_1 and x_2 in which the points $(-1, -1), (1, 1), (-3, -3), (4, 4)$ belong to one class and $(-1, 1), (1, -1), (-5, 2), (4, -8)$ belong to the other.

A set of input vectors (or a training set) will be said to be linearly non-separable if no hyperplane exists such that each vector lies on the pre-assigned side of the hyperplane.

Q.7. (b) Explain error back propagation training algorithm with the help of flowchart. (6.5)

Ans. Step 1: Normalized the inputs and outputs with respect to their maximum values. It is proved that the neutral networks work better if input and outputs lie between 0-1. For each training pair, assume there are 'l' inputs given by $\frac{(r)_I}{l \times 1}$ and 'n' outputs $\frac{(O)_o}{n \times 1}$ in a normalized form.

Step 2: Assume the number of neurons in the hidden layer to He between $l < m < 2l$

Step 3: [V] represents the weights of synapses connecting input neurons and hidden neurons and [W] represents weights of synapses connecting hidden neurons and output neurons. Initialize the weights to small random values usually from – 1 to 1. For general problems, λ can be assumed as I and the threshold values can be taken as zero.

$[V]^0 = [\text{random weights}]$

$[W]^0 = [\text{random weights}]$

$[\Delta V]^0 = [\Delta W]^0 = [0]$

Step 4: For the training data, present one set of inputs and outputs. Present the pattern to the input $(\mathrm{I})_1$ as inputs to the input layer. By using linear activation function, the output of the input layer may be evaluated as

$$\underset{l \times r}{(\mathrm{O})_1} = \underset{l \times 1}{(\mathrm{I})_r}$$

Step 5: Compute the inputs to the hidden layer by multiplying corresponding weights of synapses as

$$\underset{m \times 1}{(I)_H} = \underset{m \times l}{(V)^T} \underset{l \times 1}{(\mathrm{O})_l}$$

Step 6: Let the hidden layer units evaluate the output using the sigmoidal function as

$$\{\mathrm{O}\}_\mathrm{H} = \begin{Bmatrix} \cdot \\ \cdot \\ \dfrac{1}{(1+e^{-1_{Hi}})} \\ \cdot \\ \cdot \end{Bmatrix}_{m \times 1}$$

Step 7: Compute the inputs to the output layer by multiplying corresponding weights off synapses as

$$\underset{n \times 1}{\{I\}_o} = \underset{n \times m}{\{W\}^T} \underset{m \times 1}{(O)_H}$$

Step 8: Let the output layer units evaluate the output using sigmoidal function as

$$\{\mathrm{O}\}_\mathrm{o} = \begin{Bmatrix} \cdot \\ \cdot \\ \dfrac{1}{(1+e^{-1vj})} \\ \cdot \\ \cdot \end{Bmatrix}$$

The above is the network output,

Step 9: Calculate the error and the difference between the network output and the desired output as for the ith training set as

$$\mathrm{E}^\mathrm{P} = \frac{\sqrt{\Sigma(T_j - O_{vj})^2}}{n}$$

Step 10: Find {d} as

$$\{\mathrm{d}\} = \begin{Bmatrix} \cdot \\ \cdot \\ (\mathrm{T_k} - O_{\mathrm{ok}})\mathrm{O_{ok}}(1-\mathrm{O_{ok}}) \\ \cdot \\ \cdot \end{Bmatrix}_{n \times 1}$$

• If, a new variable z is created as function of x and y, i.e. z=f(x, y) then f assigns a particular truth value to z for each combination of truth values of x and y.

•Since n logic variables may assume 2^n prospective truth values, there are 2^{2n} possible logic funtions of these variables.

Q.8. (b) Reinforcement of learning (Ph)

Ans. It is a stochastic learning algorithm. The behave an binary units. The machine has an energy function E, given as

$$E = \frac{1}{2}\sum_{j}\sum_{k} w_{kj} \cdot x_k \, x_j$$

$E = \frac{1}{2} \sum_j \sum_k w_{kj} \cdot x_k x_j$

The states x_k or x_j are denoted by eithers 1 or – 1 w_{kj} in synaptic weight of link between j to k.j ≠ k represents a no self feedback neuron.

(i) The machine choses a neuron at random.

(ii) It then flips the state of neuron k from x_k to $-x_k$ with probability.

Where ΔE_k is change in energy function of machine T is temperature.

(iii) This rule is applied repeatedly till a thermal equilithium is attained. These are two types of neurons:

(a) Visible neurons are interface between network and operating environment,

(b) Hidden neurons operate freely.

There are two states of operation of neurons:

(a) Clamped: When invisible neurons are all clamped into specific states (r_{kj}^{+}).

(b) Free running condition: Where all neuron (visible/hidden) freelly (r_{kj}^{-}). According to Boltzmanu rule:

$$\Delta\omega_{kj} = \eta(\rho^{+}_{kj} - \rho^{-}_{kj}) \text{ (j ≠ k i.e., no self loop)}$$

Where, ρ_{kj}^{+} correlation between states of neurons j and k in the free-running condition of network.

ρ_{kj}^{+} and ρ_{kj}^{-} between – to + 1.

And η is learning rate.

Q.8. (c) Associative memory (Ph)

Ans. Refer Q.1 (g) of First Term Examination 2017.

Q.8. (d) Fitness functions

Ans. The fitness function simply defined is a function which takes a candidate solution to the problem as input and produces as output how "fit" our how "good" the solution is with respect to the problem in consideration.

Calculation of fitness value is done repeatedly in a GA and therefore it should be sufficiently fast. A slow computation of the fitness value can adversely affect a GA and make it exceptionally slow.

A fitness function should possess the following characteristics –

•The fitness function should be sufficiently fast to compute.

•It must quantitatively measure how fit a given solution is or how fit individuals can be produced from the given solution.

•In some cases, calculating the fitness function directly might not be possible due to the inherent complexities of the problem at hand. In such cases, we do fitness approximation to suit our needs.

• The following image shows the fitness calculation for a solution of the 0/1 Knapsack. It is a simple fitness function which just sums the profit values of the items being picked (which have a 1), scanning the elements from left to right till the knapsack is full.

0	1	2	3	4	5	6	Item Number
0	1	0	1	1	0	1	Chromosome
2	9	8	5	4	0	2	Profit Values
7	5	3	1	5	9	8	Weight Values

Knapsack capacity = 15

Total associated profit = 18

Last item no picked as it exceeds knapsack capacity.

9:30