NAME : ANURAN CHAKRABORTY
ROLL : 001610501020 REG. NO : 135726
DATE : 27/06/2020 MOB. NO. : 8902689001
EMAIL : ch.anuraan@gmail.com

Q1. The idea of soft computing was initiated by Lofti A. Zadeh.
Softcomputing 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 components are fuzzy logic, neuro computing, 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. Zadeh defines SC into one multidisciplinary system as the fusion of the fields of Fuzzy Logic, Neuro-computing, Genetic computing and probabilistic computing ~~Fa~~ i.e. fusion of methodologies designed to model and enable solutions to real world problems, which are not modelled or too difficult to model mathematically.

Anuran Chakraborty.

Roll : 001610501020

The guiding principle of 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
  - Uncertainty
  - Imprecision.

| Soft Computing | Hard Computing |
| --- | --- |
| • Soft computing is liberal of inexactness, uncertainty, partial truth, and approximation. | • Hard computing needs an exactly state analytic model. |
| • Produces approximate results | • Produces precise results. |
| • Will ~~em~~ emerge its own programs. | • Requires programs to be written. |
| • Incorporates randomness | • Hard computing is settled. |
| • Used multivalued logic | • uses two-valued logic |
| • Works on ambiguous and noisy data. | • Works on exact data. |
| • Stochastic in nature | • Deterministic in nature. |
| • Relies on formal logic and probabilistic reasoning | • Relies on binary logic and crisp system. |

Anuran Chakraborty

Roll: 001610501020

Ans.

| 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 algorithm 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 to learn more about the data. |
| • It is called so because the process of 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 class and class labels known. | • Either no of class, or class labels or both are unknown. |
| • Ex: SVM, Neural Networks, etc. | • Ex: K-means clustering, etc. |

Anuran Chakraborty

Roll: 001610501020

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 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 table is. Without non-linearity the neural network would fail to learn such complex mappings.

Anuran Chakraborty

Roll : 001610501020

Competitive learning is a form of unsupervised learning in artificial neural networks, which nodes compete for the right to respond to a subset of the input data.. It is a variant of Hebbian Learning, competitive learning works by increasing the specialization of each node in the network. It is well suited to finding clusters within data.

In this learning: ~~during~~

→ During training, the output unit that provides the highest activation to a given input pattern is declared the weights of the winner and is moved closer to the input pattern, whereas the rest of the neurons are left unchanged.

→ The strategy is also called winner-take-all since only the winning neuron is updated.

→ Output units may have lateral inhibitory connections so that a winner neuron can inhibit others by an amount proportional to its activation levels.

The elements of a competitive learning rule is:

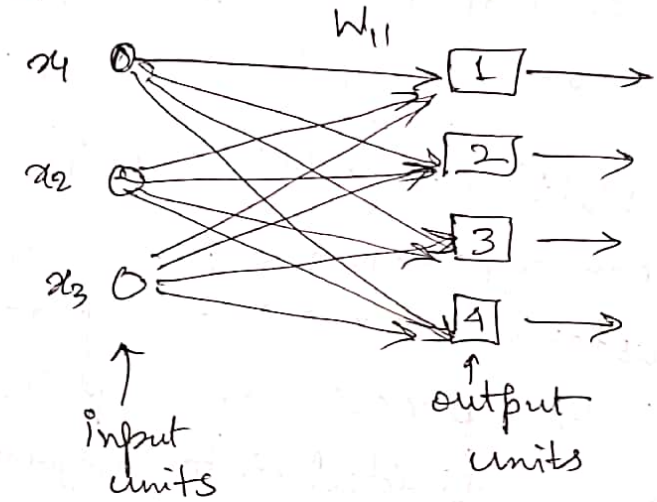
- A set of neurons that are all same except for some randomly distributed synaptic weights, and which therefore respond differently to a given set of inputs
- A limit imposed on the strength of each neuron.
- A mechanism that permits the neurons to compete for the right to respond to given subset of inputs, such that only one neuron wins.

Anuran Chakraborty

Scanned with CamScanner

with no available information regarding the desired outputs, unsuperwised learning networks update weights only on the basis of input patters. The competitive learning network is a popular scheme.

Ex.




The input vector $x = [x_1\ x_2\ x_3]^T$ and the weight vector $w_j = [w_{1j},\ w_{2j}\ w_{3j}]^T$ for an output unit $j$ are generally assumed to be normalized to unit length. The activation value $a_j$ of output unit $j$ is then calculated by the inner product of the input and weight vectors.

$$a_j = \sum_{i=1}^{3} x_i w_{ij} = x^T w_j = w_j^T x$$

Next, the output with the highest activation must be selected for further processing. Say unit $k$ has max activation the weights are updated according to winner-take-all learning rule.

$$w_k(t+1) = \frac{w_k(t) + \eta(x(t) - w_k(t))}{\|w_k(t) + \eta(x(t) - w_k(t))\|}$$

Anuran Chakraborty

Roll: 001610501020

Using Euclidean distance as a dissimilarity measure is a more common scheme, in which activation of output unit $j$

$$a_j = \left(\sum_{i=1}^{3}(x_i - w_{ij})^2\right)^{0.5} = \|x - w_j\|$$

The weights of the output unit with the smallest activation are updated according to

$$w_k(t+1) = w_k(t) + \eta\,(x(t) - w_k(t))$$

A competitive learning network performs an on-line clustering process on input patterns. When the process is complete, the input data are divided into disjoint clusters such that similarities between individuals in the same cluster are larger than those in different clusters. However, it lacks the capability to add new clusters when deemed necessary.

Anuran Chakraborty

Q2.

$$R = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.9 & 1 & 0.1 \end{bmatrix} \qquad S = \begin{bmatrix} 0.5 & 0.8 \\ 0.1 & 1 \\ 0 & 0.6 \end{bmatrix}$$

## RoS as Max-product composition

~~RoS~~

Max-product composition can be defined as

$$\mu_{R_1 o R_2}(x,z) = \max_y \left[\mu_{R_1}(x,y) \cdot \mu_{R_2}(y,z)\right]$$

If matrices are used then the operation is similar to matrix multiplication except we use max operation instead of '+'.

$$RoS = \begin{bmatrix} \max(0.4\times0.5,\ 0.6\times0.1,\ 0) & \max(0.4\times0.8,\ 0.6\times1,\ 0\times0.6) \\ \max(0.9\times0.5,\ 1\times0.1,\ 0.1\times0) & \max(0.9\times0.8,\ 1\times1,\ 0.1\times0.6) \end{bmatrix}$$

$$= \begin{bmatrix} \max(0.2,\ 0.06,\ 0) & \max(0.32,\ 0.6,\ 0) \\ \max(0.45,\ 0.1,\ 0) & \max(0.72, 1,\ 0.06) \end{bmatrix}$$

$$= \begin{bmatrix} 0.2 & 0.6 \\ 0.45 & 1.0 \end{bmatrix}$$

Anuran Chakraborty

Roll: 001610501020

$$R = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0.9 & 1 & 0.1 \end{bmatrix} \qquad S = \begin{bmatrix} 0.5 & 0.8 \\ 0.1 & 1 \\ 0 & 0.6 \end{bmatrix}$$

**RoS as ~~min-max~~ max-min composition:**

The max-min composition of $R_1, R_2$ is defined by

$$R_1 \circ R_2 = \left\{ \left[ (x,z), \max_y \min\left(\mu_{R_1}(x,y), \mu_{R_2}(y,z)\right)\right], \; x \in X, \; y \in Y, \; z \in Z \right\}.$$

or

$$\mu_{R_1 \circ R_2}(x,z) = \max_y \min\left[\mu_{R_1}(x,y), \mu_{R_2}(y,z)\right]$$

$$R \circ S = \begin{bmatrix} \max(\min(0.4,0.5), \min(0.6,0.1), \min(0,0)) & \max(\min(0.4,0.8), \min(0.6,1), \min(0,0.6)) \\ \max(\min(0.9,0.5), \min(1,0.1), \min(0.1,0)) & \max(\min(0.9,0.8), \min(1,1), \min(0.1,0.6)) \end{bmatrix}$$

$$= \begin{bmatrix} \max(0.4,0.1,0) & \max(0.4,0.6,0) \\ \max(0.5,0.1,0) & \max(0.8,1,0.1) \end{bmatrix}$$

$$= \begin{bmatrix} 0.4 & 0.6 \\ 0.5 & 1.0 \end{bmatrix}$$

Anuran Chakraborty

Roll : 001610501020

Four frequently used T-norm operators :

- Suppose $R: A \rightarrow B = A \times B = \int_{X \times Y} \mu_A(x) * \mu_B(y) \big|_{(x,y)}$

where $*$ is called a t-norm operator.

t-norm or triangular norm is a kind of binary operation used in fuzzy logic. A t-norm generalises intersection in a lattice and conjuction in logic. A t-norm is a function $T: [0,1] \times [0,1] \rightarrow [0,1]$ which satisfies

- Commutativity : $T(a,b) = T(b,a)$
- Monotonicity : $T(a,b) \leq T(c,d)$ if $a \leq c$ and $b \leq d$
- Associativity : $T(a, T(b,c)) = T(T(a,b), c)$
- 1 is identity element : $T(a,1) = a$

Most frequently used T-norm operators are :

- minimum : $T_{min}(a,b) = min(a,b) = a \wedge b$.
- Algebraic Product : $T_{ap}(a,b) = ab$

Product t-norm is the standard semantics for strong conjunction in product fuzzy logic.

- Bounded product : $T_{bp}(a,b) = 0 \vee (a+b-1)$
- Drastic product : $T_{dp} = \begin{cases} a & \text{if } b = 1 \\ b & \text{if } a = 1 \\ 0 & \text{if } a,b < 1 \end{cases}$

Here $a = \mu_A(x)$ and $b = \mu_B(y)$. and $T_*$ is called the function of T-norm operator.

Anuran Chakraborty

Roll: 001610501020

$$\mu_{TALL}(\text{height}) = \left\{ \frac{0.5}{5'}, \frac{0.8}{6'}, \frac{1.0}{7'} \right\}$$

$$\mu_{MODERATE}(\text{weight}) = \left\{ \frac{0.7}{45kg}, \frac{0.9}{50kg} \right\}$$

$$\mu_{HIGH}(\text{speed}) = \left\{ \frac{0.6}{5ms^{-1}}, \frac{0.7}{10ms^{-1}}, \frac{0.9}{15ms^{-1}} \right\}$$

$$\mu_{MORE-OR-LESS-TALL}(\text{height}) = \left\{ \frac{0.7}{5'}, \frac{0.8}{6'}, \frac{0.6}{7'} \right\}$$

$$\mu_{MORE-OR-LESS-MODERATE}(\text{weight}) = \left\{ \frac{0.75}{45kg}, \frac{0.85}{50kg} \right\}$$

find $\mu_{MORE-OR-LESS-HIGH}(\text{speed})$

Fuzzy production rule is
"IF height is TALL and weight is MODERATE,
THEN speed is HIGH."

There are two antecedants connected by AND $\therefore$ to get antecedant membership function $\mu_{AM}(\text{height, weight}) = t(\mu_{TALL}(\text{height}), \mu_{MODERATE}(\text{weight}))$

$t$ is the t-norm.

according of Mamdani implication

$$t(x,y) = \min(\mu_A(x), \mu_B(y))$$

i.e. the union

$$\mu_{TALL}(\text{height}) = \left\{ \frac{0.5}{5'}, \frac{0.8}{6'}, \frac{1.0}{7'} \right\}$$

$$\mu_{MODERATE}(\text{weight}) = \left\{ \frac{0.7}{45kg}, \frac{0.9}{50kg} \right\}.$$

Anuran Chakraborty

Roll : 001610501020

$$\mu_{AM}(\text{height, weight}) = \left\{ \frac{\min(0.5, 0.7)}{5', 45kg}, \frac{\min(0.5, 0.9)}{5', 50kg}, \frac{\min(0.8, 0.7)}{6', 45kg}, \frac{\min(0.8, 0.9)}{6', 50kg}, \frac{\min(1.0, 0.7)}{7', 45kg}, \frac{\min(1.0, 0.9)}{7', 50kg} \right\}$$

$$= \left\{ \frac{0.5}{5', 45kg}, \frac{0.5}{5', 50kg}, \frac{0.7}{6', 45kg}, \frac{0.8}{6', 50kg}, \frac{0.7}{7', 45kg}, \frac{0.9}{7', 50kg} \right\}$$

- The second phase is computation of the implication by Lukasiewicz function.

Let us represent the relation by L.

$$\mu_L(x, y, z) = \min(1, (1 - \mu_{AM}(x, y) + \mu_C(z)))$$

Converting the relations to matrix form.

$$\mu_L(\text{height, weight, speed}) =$$

$$\begin{array}{c|c} & \begin{array}{ccc} 5ms^{-1} & 10ms^{-1} & 15ms^{-1} \end{array} \\ \begin{array}{c} 5', 45kg \\ 5', 50kg \\ 6', 45kg \\ 6', 50kg \\ 7', 45kg \\ 7', 50kg \end{array} & \begin{bmatrix} 1.0 & 1.0 & 1.0 \\ 1.0 & 1.0 & 1.0 \\ 0.9 & 1.0 & 1.0 \\ 0.8 & 0.9 & 1.0 \\ 0.9 & 1.0 & 1.0 \\ 0.7 & 0.8 & 1.0 \end{bmatrix} \end{array}$$

$$\mu_L(5', 45kg, 5ms^{-1}) = \min(1, (1 - 0.5 + 0.6))$$
$$= 1.0.$$

Anuran Chakraborty

Roll : 001610501020

To find the membership distribution of the fuzzy inference : speed is MORE-OR-LESS-HIGH. To find it we can do ~~the following~~ compose the t-norm $t\left(\mu_{\text{MORE-OR-LESS-TALL}}(\text{height}),\ \mu_{\text{MORE-OR-LESS-MODERATE}}(\text{weight})\right)$

with $\mu_L(\text{height, weight, speed})$.

$$\mu_{\text{MORE-OR-LESS-HIGH}}(\text{speed}) = t\left(\mu_{\text{MORE-OR-LESS-TALL}}(\text{height}), \mu_{\text{MORE-OR-LESS-MODERATE}}(\text{weight})\right) \circ \mu_L(\text{height, weight, speed})$$

~~t(x,y)~~ Using the previous definition of t-norm.

$$t\left(\mu_{\text{MORE-OR-LESS-TALL}}(\text{height}), \mu_{\text{MORE-OR-LESS-MODERATE}}(\text{weight})\right)$$

$$= \left\{ \frac{\min(0.7, 0.75)}{5', 45kg}, \frac{\min(0.7, 0.85)}{5', 50kg}, \frac{\min(0.8, 0.75)}{6', 45kg}, \frac{\min(0.8, 0.85)}{6', 50kg}, \frac{\min(0.6, 0.75)}{7', 45kg}, \frac{\min(0.6, 0.85)}{7', 50kg} \right\}.$$

$$= \left\{ \frac{0.7}{5', 45kg}, \frac{0.7}{5', 50kg}, \frac{0.75}{6', 45kg}, \frac{0.8}{6', 50kg}, \frac{0.6}{7', 45kg}, \frac{0.6}{7', 50kg} \right\} = \mu'_{AM}$$

Anurran Chakraborty

Composing the above result                                  Roll: 001610501020

$$\cancel{\begin{array}{c|cc} & 45kg & 50kg \\ \hline 5' & 0.7 & 0.7 \\ 6' & 0.75 & 0.8 \\ 7' & 0.6 & 0.6 \end{array}} \quad \cancel{0}$$

using max-min composition

$$\mu_{MOL-HIGH} = \mu'_{AM} \circ \mu_L$$

$$\begin{array}{c|ccc} & 5ms^{-1} & 10ms^{-1} & 15ms^{-1} \\ \hline 5',45kg & 1.0 & 1.0 & 1.0 \\ 5',50kg & 1.0 & 1.0 & 1.0 \\ 6',45kg & 0.9 & 1.0 & 1.0 \\ 6',50kg & 0.8 & 0.9 & 1.0 \\ 7',45kg & 0.9 & 1.0 & 1.0 \\ 7',50kg & 0.7 & 0.8 & 1.0 \end{array} \circ \begin{array}{c|cc} & 45kg & 50kg \\ \hline 5' & 0.7 & 0.7 \\ 6' & 0.75 & 0.8 \\ 7' & 0.6 & 0.6 \end{array} \; ?$$

$$\mu_{MORE-OR-LESS-HIGH}(5\,ms^{-1})$$
$$= \max\big(\min(\mu'_{AM}(5',45kg), \mu_L(5',45kg,5ms^{-1})),$$
$$\min(\mu'_{AM}(5',50kg), \mu_L(5',50kg,5ms^{-1})),$$
$$\min(\mu'_{AM}(6',45kg), \mu_L(6',45kg,5ms^{-1}))$$
$$\ldots \min(\mu'_{AM}(7',50kg), \mu_L(7',50kg,5ms^{-1}))\big)$$

$$\cancel{= \max(\min(1,0.7), \min(1,0.7))}$$

$$= \max(\min(0.7,1), \min(0.7,1), \min(0.75,0.9), \min(0.8,0.8), \min(0.6,0.7), \min(0.6,0.7))$$

$$= \max(0.7, 0.7, 0.75, 0.8, 0.6, 0.6) = 0.8$$

Anwesan Chakraborty

Roll: 001610501020

$$\mu_{MORE-OR-LESS-HIGH}(10ms^{-1}) = \max\{\min(0.7,1), \min(0.7,1), \min(0.75,1), \min(0.8,0.9), \min(0.6,1), \min(0.6,0.8)\}$$

$$= \max(0.7, 0.7, 0.75, 0.8, 0.6, 0.6)$$

$$= 0.8$$

$$\mu_{MORE-OR-LESS-HIGH}(15ms^{-1}) = \max(\min(0.7,1), \min(0.7,1), \min(0.75,1), \min(0.8,1), \min(0.6,1), \min(0.6,1))$$

$$= \max(0.7, 0.7, 0.75, 0.8, 0.6, 0.6)$$

$$= 0.8$$

$$\therefore \mu_{MORE-OR-LESS-HIGH}(speed) = \left\{ \frac{0.8}{5ms^{-1}}, \frac{0.8}{10ms^{-1}}, \frac{0.8}{15ms^{-1}} \right\}$$

Anwran Chakraborty.