SOFT COMPUTING

SUBJECT CODE:18KP3CS10

Department of Computer Science

A.Thirumalairaj
N. Babykala
G. Umarani

SOFT COMPUTING

SUBJECT CODE 18KP3CS10

UNIT-III-Backpropagation Networks(BPN):Architeture of BPN-Backpropagationlearning -Applications-Effect of Tuning parameters in BPN- Selection of various parameter in BPN-Variations of Standard Backprobgation Alogorthim –Research Direction.

UINT–IV-Fuzzy Set Theory –Crips Sets-Fuzzy Sets –Crips Releations-Fuzzy Relations-Fuzzy Systems:Crisp Logic-Predicate Logic-Fuzzy Logic-Fuzzy Rule based system-Defuzzilification Methods-Applications.

UNIT –V.Genetic Algorthim(GA):Fundamental-Basic Concepts-Creation offspring-Working Principle-Encoding-Fitness Function-Reproduction-Genetic Modeling-Inheritance operator-Cross over-Inversion and Deletion-Mutation Operator-Bitwise Operators-Generational Cycle-Convergence of GA-Application. Book:

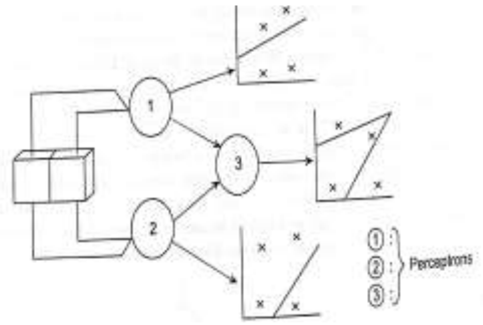
Reference Book

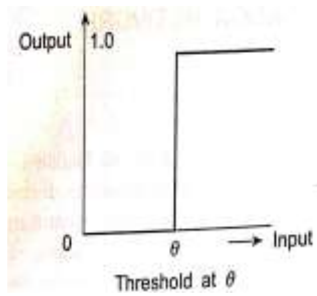
Neural Networks,Fuzzy Logic and Genetic Algorthims and Synthesis and Application-S.Rajasekaran-G.A.Vijayalakshmi Pai

UNIT-3 BACK PROPAGATION NETWORK

Architecture of Back Propagation Network

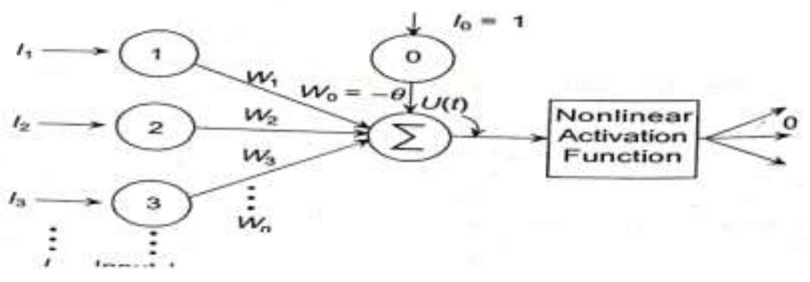
- Combing preceptrons to solve XOR Problem
- Each neuron in the structure take s weighted sum of inputs,theroshold it and output either a one or zero
- For the perceptron in the first layer ,the input comes from the actual inputs of the problem
- The perceptrons of the second layer do not know which of the real inputs from the first layer were on or off.
- The actual inputs are effectively masked off from the output units by any indication scale by to adjust the weights.
- The two states of neuron being on or off ,any indication on the weights
- The hard hitting thershold factions remove the information that is needed if the network is successfully learn.
- Thresholding process is to adjust it slightly and use to slightly different nonlinearity.
-





The Solution

- To Strengthen or Weaken the relevant weights
- A couple of possibilities for the new thresholding function
- In case the input and threshold are almost Same the output of the neuron value between 0 and 1
- The output to the neurons can be related to input.
- Analogous to biological neuron, an artificial neuron received much input representing the output of the other neurons
- Each input is mutiliplied by the corresponding weights analogs to the synaptic strengths.
- All of these weighted inputs are summed up and passed though an activation function to determine the neuron.



-
- $U(t)=W_1I_1 + W_2I_2 +\ldots.+W_nI_n$ or $u = \langle W \rangle \{I\}$

Considering thershold Θ, the relative input to the neuron is given by

$u(t)=W_1I_1 + W_2I_2 +\ldots..+ W_nI_n - \Theta$

$= \sum_{i=0}^{n} = W_iI_i$ where $W_0=-\Theta$; $I_0=1$

The ouput using the nonlinear transfer function ‘f’ is give m

$0=f(u)$ The current mechanism biological neuron

Single Layer Artificial Neural Network

- Consider a single layer feed forward neural network consisting of an input layer to receive the inputs an output layer to output the vectors
- The input layer consits of n neuron and the output layer m neurons
- i th neroun ,jth ouput neuron as W_{ij}

$$\begin{Bmatrix} I_1 \\ I_2 \\ \\ I_n \end{Bmatrix}_{n\,x1} \quad o., \quad \begin{Bmatrix} O_{01} \\ O_{02} \\ \\ O_{03} \end{Bmatrix}_{mx1}$$

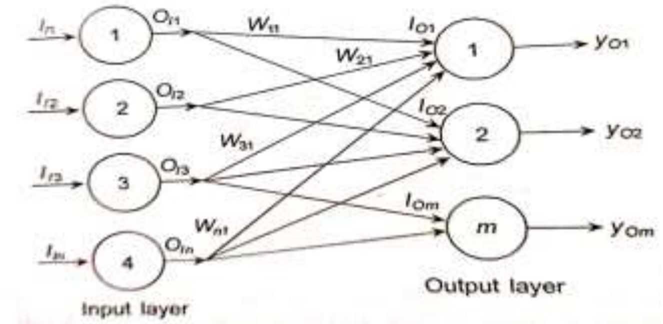
Sigmoidal function for the neurons in the ouput layer

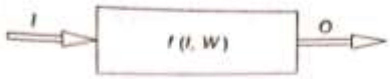
{O1}={I1} (linear transfer function)

n x1 mx1. $IOJ=W_{1J}I_{I1} + W_2JI_{12} +\ldots..+WnjI_{IN}$

Hence the input to the output layer

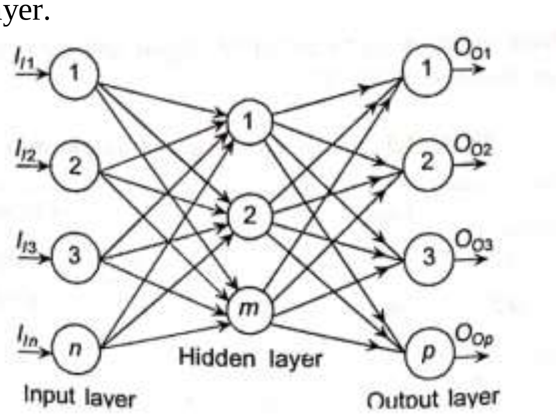
$\{I_O\} =[W]^T \{O_1\}=[W]^T\{I_1\}$

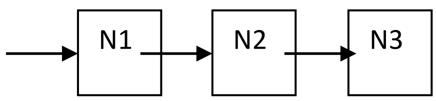




Model for Layer Perceptron

- This model Layer three layer an Input layer,and ouput layer and a layer in between connected directly to the input or the ouput called hidden Layer
- Linear function used in hidden layer and output layer
- Using Sigmoidal function or squashed function –s Function
- The input layer recevices the values next layer .Using the sigmoidal function added the hidden layer
- The mapping of mulilayer perceptron $O=N_3[N_2\ [N_1[I]]]$
- N1 and N2 and N3 represent non linear mapping provided input hidden and ouput layer
- Mulitilayer perceptron provides no increases in computational power over single layer neural networks
- The activity of neuron in the input layers represents the raw information that fed into the network
- The activity of neurons in the hidden layer is determined activities of the neurons in the input layer and the connecting weights between input and hidden units.
- The activity of the output units depend on the activity neurons in the hidden Layer and the ouput Layer.





-
-

BACKPROPAGATION LEARNING

- Input layer Computation
- Hidden Layer Computation
- Output layer Computation
- Calculation of Error
- Training of Neural Network

Input Layer Computation:

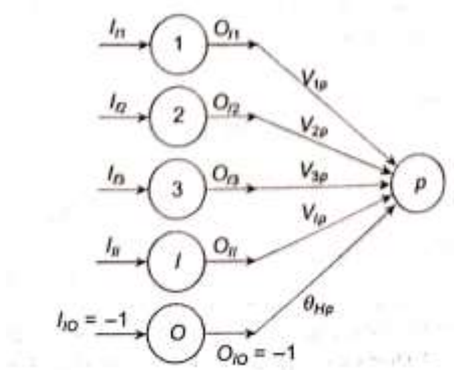
- To use linear activation function the output of the input layer
- One set of data $\{o\}_I=\{1\}_I$,l x1, l x 1
- The hidden neurons are connected sysnapes to input neuron v_{ij} is the weight of the arc between
- I th input neuron to jth hidden neuron is the weighted sum of the ouputs of the input neuron to get I_{HP}
- $I_{HP} = V_{1P}O_{I1} + VO_{I1} + V_{2P} O_{I2} + \ldots\ldots\ldots.. V_{IP} O_{il}$ (p=1,2,3…m)
- Denoting weight matrix or Connectivity matrix between input neurons and hidden neurons [v]
- We get the an input to the hidden neurons as
- $\{I\}_H=[V]^T \{O\}_I$, m x 1 mx l lx 1
-

Hidden Layer Computation

Sigmoidal function or squashed –s function the output of the pth hidden neuron

$$O_{Hp} = \frac{1}{(1 + e^{-\lambda(I_{Hp} - \theta_{Hp})})}$$

O_{HP} is the output of the hidden neurons ϴhp is the thershold of the pth neuron.A non-zero threshold neuron is computationally equivalent to an input that is always held at -1 and non-zero becomes the Connecting weight value



Treating threshold in hidden layer

our dervations we will not treat thershold

$$\{O\}_H = \left\{\begin{matrix} - \\ - \\ \frac{1}{(1 + e^{-\lambda(I_{Hp} - \theta_{Hp})})} \\ - \\ - \end{matrix}\right\}$$

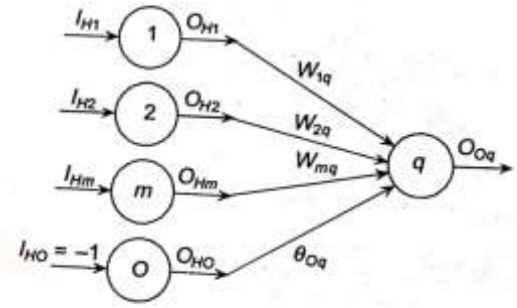
- Output of the hidden neuron to get l_{og}
- $I_{OQ}=W_{1Q}O_{H1} + W_{2Q}O_{H2}+\ldots\ldots\ldots.+W_{mq} O_{hm}$ (q1,2,3….n)
- Input to the output neuron as
- $\{I\}_O =[W]_T \{O\}_{H.}$., n x1 n x m mx1

Output Layer Computation

Sigmoidal function the output of the qth output neuron

$$O_{Oq} = \frac{1}{(1 + e^{-\lambda(I_{Oq} - \theta_{Oq})})}$$

O oq is the output of the qth output neuron ,Ioq is the input to the qth output neuron and Θ_{oq} is the thershold of the qth neuron.Oth neuron in the hidden layer with output of -1 and the threshold value Ɵoq becomes the connecting value



$$\{O\}_O = \begin{Bmatrix} - \\ - \\ \frac{1}{(1+e^{-\lambda(I_{Oq}-\theta_{Oq})})} \\ - \\ - \end{Bmatrix}$$ Output of neuron

Calculation of Error

Rth output neuron and for the traning example 'O' for the target output 'T'

$$E_r^1 = \frac{1}{2} e_r^2 = \frac{1}{2}(T - O)^2$$

Where E_r^1 ==1/2 second norm of the errror in the rth neuron for the given traning pattern.

The Square of the error is postive or negative .The Eculdean norm of error E^1 for the first traning pattern

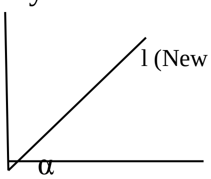
$$E^1 = \frac{1}{2}\sum_{r=1}^{n}(T_{Or} - O_{Or})^2$$

gives the error fuction in one traning pattern,same techniques for all training pattern

$$E(V,W) = \sum_{j=1}^{nset} E^j(V,W,I)$$

- E is the error function depending on the m(1+n) weights of [w] and [v]
- Defined to be maxized or minimized with respect to set of parameters
- The network parameters the optimize the error fuction E over the nset of pattern $[I^{nset}, t^{nset}]$
- [v] and [w]
- l x m m x n

Training of Neural network

- The similarity messure between the input vector I and the synaptic weights [V] and [W]
- The Neural Network learns this new pattern, by changing the strenghts of synaptic and weights
- Similarity between new information and past knowledge
- 

Weights are Wand V are adjusted .

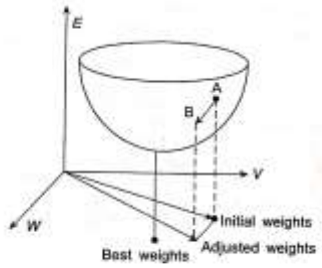
Algorithm for determing the connection strength to ensure learnings are called Learning rules

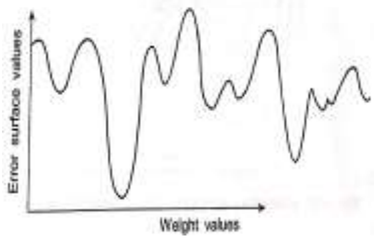
Method of Steepest Descent

Calculation of error surface is given by

E=$\sum_{p=1}^{n\ Set}$ E^p(V,W,I)

Multilayer feedforward networks with non linear activation functions have mean squared error surface Above the total Q-dimensional weight space R^Q



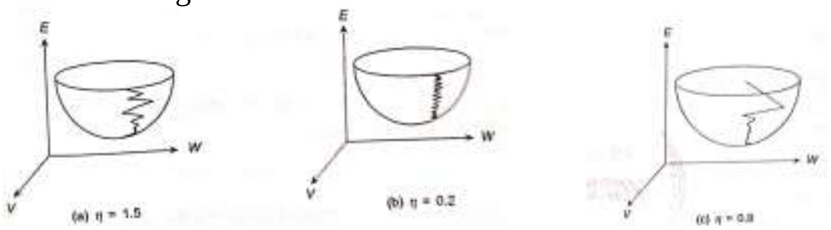


At given error value *E* including minima regions there are many permuation of weights rise same value of E.BP assured of finding global minimum as the simple layer delta rule
The learning process on the flat or near flat region of the error surface
At the traning process gradient process,location with error value E determined initial weight assignments W(0),V(0) and the traning pattern pair (I^P,O^P)

$$E = \frac{1}{\text{nset}} \sum_{p=1}^{\text{nset}} E^p = \frac{1}{2 \times \text{nset}} \sum_{p=1}^{k} (T_k^p - O_{Ok}^p)^2$$

Effect of Learning Rate ′η′

The learning coefficient is too small the descent will progress in small steps significantly increasing the Time to converge. example the learning coefficient is taken as 0.9 and this seems to optimistic the value Of the learning coefficient function of error derivate on successive updates



Covergence paths for different learning coefficents

Adding a Momentum Term

To improve the rate of convergence by adding some inertial momentum to gradient expression.
This can be accomplished by adding a fraction of the previous weight change to the current weight Change. The addition of such term smooth out the descent path by preventing extreme changes in the Gradients due to local anamalies.

$[\Delta W]^{t+1} = -'\eta' = \frac{\partial E}{\partial W} + \alpha[\Delta W]^t$
$[\Delta V]^{t+1} = -'\eta' = \frac{\partial E}{\partial W} + \alpha[\Delta V]^t$
α is defined as the moments coefficient. The value of α but positive less than 1.value lie in the range 0.5-0.9

$$[\Delta W]^{t+1} = \eta\{O\}_H \langle d \rangle + \alpha [\Delta W]^t$$

$$[\Delta V]^{t+1} = \eta\{I\}_I \langle d^* \rangle + \alpha [\Delta V]^t$$

The threshold value

$$\{\Delta\theta\}_O^{t+1} = \eta\{d\} + \alpha\{\Delta\theta\}_O^t$$

$$\{\Delta\theta\}_H^{t+1} = \eta\{d^*\} + \alpha\{\Delta\theta\}_H^t$$

Weights and threshold may be updated as

$$[W]^{t+1} = [W]^t + [\Delta W]^{t+1}$$

$$[V]^{t+1} = [V]^t + [\Delta V]^{t+1}$$

$$[\theta]_O^{t+1} = [\theta]_O^t + [\Delta\theta]_O^{t+1}$$

$$[\theta]_H^{t+1} = [\theta]_H^t + [\Delta\theta]_H^{t+1}$$

Backpropagation Algorthim

The learning **algorithm** for to develop the multilayer feedforward neural network .To minimize the error . To Consider three layer input layer I nodes,hidden layer m nodes,and Output layer n nodes **.**To using sigmoidal function for activation function for the hidden
and output layers and linear activation function used for input layer.To choose the hidden layer to chosen to l and 2l.Basic Algorithm loop structure.

Initialize the weights

Repeat

For each training pattern

Train on that pattern

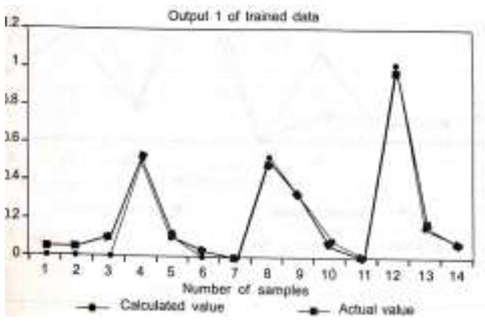
End

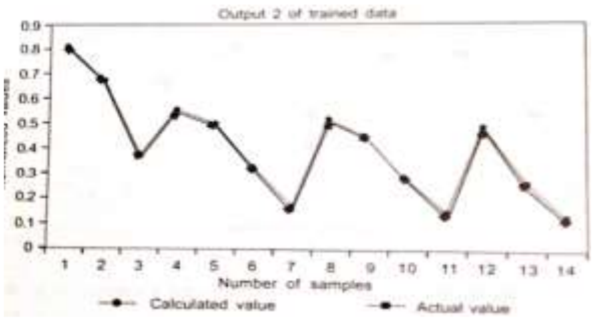
Until the error is acceptably low

APPLICATIONS

Design of journal Bearning

The Design of Bearing depend on the load,speed of the journal clearance in the bearning ,length and diameter of the bearning and the kind of Surface.
NEURONET is used to train the data and infer the result for test data.A backpropagation neural network with 8 input neurons,8 hidden neurons and 2 output neurons .5000 iteration have been performed till the error rate converges to the tolerance. The traning data and testing data inputs and outputs in rane of 0-1





Classification of Soil

To used that construction of building the soil to classified sand ,gravel or many other to mixture the varying Proportions of particles of different size. The neural network process to applied to grouping the soil 6 input neurons ,6 hidden neurons and 1 Output neurons.The calculate the percentage of sand and gravel and other particulars W_l and plastic limit W_P.The classisfication 0.1 clayer sand (SC).0.2 for clay with medium compressibility(CI)0.3For clay low compressibility.0.6 silts with medium compressibility. The colour soil 0.1 for brown 0.2 for Brownish grey,0.3 greyiesh brown, redishh yellow 0.7 red.Thirty trained set taken as network is trained for 250 iterations .Learning rate of monument value 0.6 and 0.9.The error rate of 250^{th} iteration found to be 0.0121.The rejection rate was found to be nil for trained set and 8% for untrained data set.

Hot Extrusion of Steel

Hot metal forming is important process in industry to achieve the energy and material saving quality **Improvement and de**velopment.The simulation and analysis process forging head,equivalent stress and equivalent stress and equivalent strain simulated with a finite element code for C45 Steel billet.
The neural nertwork process applied the process to three layer 2 input neurons,8 neurons in hidden layer and five output neurons.The value 0 and 1.The learning momentum rate and momentum factor are take 0.6 and 0.8.The error rate reach the tolerance value at 3500 itrations.The inputs are die angle and punch velocity and the outputs consist of forging load,maxium equivalent strain,maxium equivalent strees,

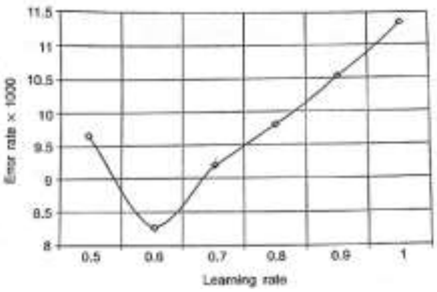
maxium equivalent velocity,and equivalent strain rate.24 data sets are taken for training and eighteen data sets are taken for testing.

EFFECT OF TUNING PARAMETERS OF THE BACKPROPAGATION NEURAL NETWORK

The Proper selection of tuning parameters for the backpropagation neural network to achieve the stable to learning the Coefficient ,Sigmoidal gain and threshold value. Weight adjustment is made based on the momentum method The range of learning coefficient that will produce rapid training depends on the number and type of input pattern.To selecting the empirical formula

$$\eta = \frac{1.5}{(N_1^2 + N_2^2 + \cdots + N_m^2)}$$

N_1 is the number of patterns type 1 and m is the number of different pattern types. Small value of leaning Coefficient produce the slower but stable training. The largest value of leaning coefficient is separate type.The learning coefficient is large is greater than 0.5 the weights are changed drastically. if the learning is small that is less than 0.2 the weights are changed in small increment. The learning rate has chosen as high as possible to allow fast learning without leading to oscillation. The learning rate and error rare for soil mechanics optimum learining rate is 0.6



Sigmoidal gain

sigmoidal function is selected, the input-output relationship of the neuron can be set as

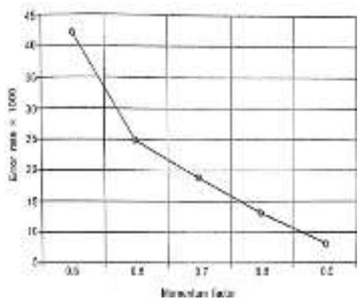
$$O = \frac{1}{(1 + e^{-\lambda(I+\theta)})}$$

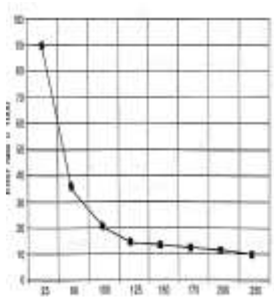
$\lambda$is a scaling factor known as sigmoidal gain. the input-output characteristic of the analog neuron approaches that the two-state neuron or the activation function approaches the Sat function.

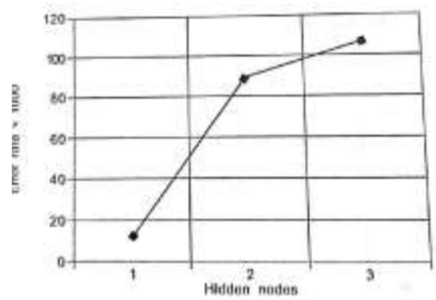
The value of sigmoidal gain also affects backpropagation. Improper combinations of the scaling factor, learning rate, and momentum factor might lead to over correction and poor convergence.
output, as the scaling factor is increased, learning rate and momentum factor have to be decrease in order to prevent oscillations.

Threshold value

$\theta$is commonly called as threshold value of a neuron, or the bias or the noise factor.A neuron fires or generates an output if the weighted sum of the input exceeds the threshold value. choose some random values and change them during training is hard to say which method is more efficient. the variation of error rate with respect learning rate, momentum factor, number of iterations, number of hidden neurons, and the neuron of hidden layers. Let take the optimum values for the learning rate and momentum factor are taken as 0.6 and 0.9.







Hidden nodes veres error rate

SELECTION OF VARIOUS PARAMETERS IN BPN

Number of Hidden Nodes

The problem decided the number of nodes in the first and third layers. n is the number of hidden nodes.To selecting the minimum nodes to network performance the memory demand for storing the weights to be minimum.Mirchandani and Cao have proved

M is a function of the number of hidden nodes H in BPN H = M-1

H = K — 1, where K is the number of elements in the learning set.

When the number of hidden nodesto the number of training patterns, the learning could be fastest Vapnik—Chervonenkis dimension probability theory.The results also indirectly help in the selection of number of hidden nodes for a given number of training patterns VCdim for BPN is given by its number of weights which is equal to 11#12 + 12*13, where 11 and 13 denote input and output nodes and 12 denotes hidden nodes. training samples T to be greater than VCdim.

$$10*T = \frac{l_2}{(l_1 + l_3)}$$

$$l_2 = \frac{10T}{(l_1 + l_3)}$$

Momentum Coefficient α

To reducing the time is use of momentum factor enhance the training phase

$$[\Delta W]^{n+1} = -\eta \frac{\partial E}{\partial W} + \alpha [\Delta W]^n$$

The use of momentum term carry a weight changes process through the one local minima and get it into global minima. Only disadvantage is cost effective compared to other Method main drawback of the BPN is the long and sometimes uncertain training time.

Sigmoidal Gain λ

Sigmoidal function is falt,a better method of coping with network paralysis is to adjust the sigmoidal Gain.Sigmoidal function on wide range training proceeds faster.

Local Minimal

All weights by specific or random amounts. If this fails, then the most practical solution reRandom the weights and start the training all over. Second method of backpropagation is used until the process seems to stall. Simulated annealing is then used to continue training until local minima has been left behind.

Learning Coefficient η

The learning coefficient cannot be negative because cause the change of weight vector to move away Ideal vector postion.the learning coefficient is Zero no learning takes place the learning coefficient must be positive.

VARIATIONS OF STANDARD BACKPROPATATION ALGORTHIM

Decremental lteration Procedure

Conventional steepest descent method, the minimum is achieved by variable

If no improvement the value of k reduced and the process is continued is decrease the error .The training of the network reach palteu error might increase leading over tuning This stage previous weights using reduced Moument factor and learning small value.

$$\Delta X = \frac{-k \frac{\partial f}{\partial X}}{\sqrt{\left\{\left(\frac{\partial f}{\partial X}\right)^2 + \left(\frac{\partial f}{\partial Y}\right)^2\right\}}}$$

Adaptive Backpropagation (Accelerated Learning)

The developed Alexander and atiya for the development of fault detection and identification system for a Process composed of direct current motor a centifigual pump and associated piping system. In ABP learning Algorthim the network weight update the error function

$$\Delta W = W(\text{new}) - W(\text{old}) - \eta \frac{\partial E}{\partial W}$$

Weight update ABP algorthim .   The learning function p(E)depends on the total errorE.

$$W(\text{new}) - W(\text{old}) - \frac{\rho(E)\left(\frac{\partial E}{\partial W}\right)}{\left\|\frac{\partial E}{\partial W}\right\|^2} \qquad \rho(E) = \begin{pmatrix} \eta \\ \eta E \\ \eta \tanh(E) \\ \eta \tanh^{-1}\left(\frac{E}{E_0}\right) \end{pmatrix}$$

Where η and E_0 are constant non negative numbers representing learning rate and the Normalization factor p(E) = ηE to provide most acceleration

$$\Delta W = \frac{-\eta E\left(\frac{\partial E}{\partial W}\right)}{\left\|\frac{\partial E}{\partial W}\right\|^2}$$

Decremental methods with adapative back propagation dependence the learning function is on the Instantaneous value of the error there by leading to in faster convergence.

Genetic Alogorithm Based Backpropagation

Coventional BPN makes use of a weight updating rules kind of gradient decent techniques to determine the weights.The robust search and optimization techniques outperforming gradient based techniques solution problems

Quick Prop Training

The most effective algorthims in overcoming the step size problem in backpropagation.A second Order method related newtons method is used to update the weights

$$E_{n+1} = E_n + \Delta W \frac{\partial E}{\partial W} + \Delta V \frac{\partial E}{\partial V} + \frac{\Delta W^2}{2} \frac{\partial^2 E}{\partial W_i \partial W_j} + \Delta W \Delta V \frac{\partial^2 E}{\partial W_i \partial V_j} + \frac{\Delta V^2}{2} \frac{\partial^2 E}{\partial V_i \partial V_j}$$

Quick props weight update procedure depends on two apporoximations-1.small changes in one weight Produce relatively little effect on the gradient error.

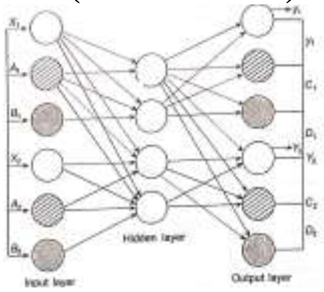
Augmented BP Networks

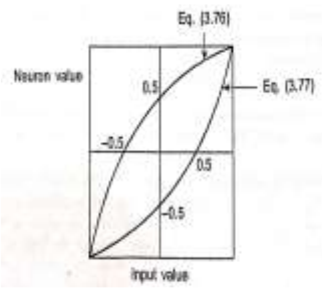
The principle of augmenting the network is

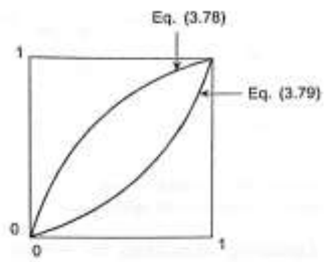
The augmented neurons are highly sensitive in the boundary domain, thereby facilitating the construction of accurate mapping 1” the model s boundary domain. 2. the network denotes each input variable with multiple input neurons, thus allowing a highly interactive functions on hidden neurons to be easily formed. The architecture of the augmented neural network is that of a standard network. The

logarithmic Neuron in the put layer receives a natural logarithm transformation of the corresponding input value traning data.

A_i =ln (1.175 X_i +1.543)







The input exponent neurons receive natural exponent tramnsformatio for corresponding input value
B_i=0.851 exp (X_i)-1.313
Where B_i is the output of the I th exponent neuron in the input layer and this transformation
The ouput layers algorithm neuron and exponent neuron transform output as
C_j=1n(1.718 Y_J + 1)
D_j=exp (0.6931 Y_J – 1)

Research Directions

New Topologies

The static and dynamic nets.Dynamic sets recurrent nets have been quit popular.To sequence of Error propagation networks input and ouput vectors are divided into internal and external portions and the network operators by concatenating the input and output vectors.

Better Learning Algorthim

A group of methods have been suggested instead of weight update on output error.Alogrithm based on some random or heuristic variations.MRII algorthim based on derivate estimation perturbations random Optimazation and techniques based on the genetic algorthim.

Better Training Startgies

It is depends model in the hidden layer,learning rate,moumentum factor,the distribution of training patternThe input space and the training algorithm.

Hardware Implementation

To design the efficient hard ware which exploits the special features of neural computing such as Parallel distributed processing.

Conscious Networks

The neural networks also contribute to the neurology and psychology. They are regulary used to model parts of living organisms and to investigate the internal mechanism of brain.the algorthim to perform the specific task to research.

UNIT IV: FUZZY SET THEORY

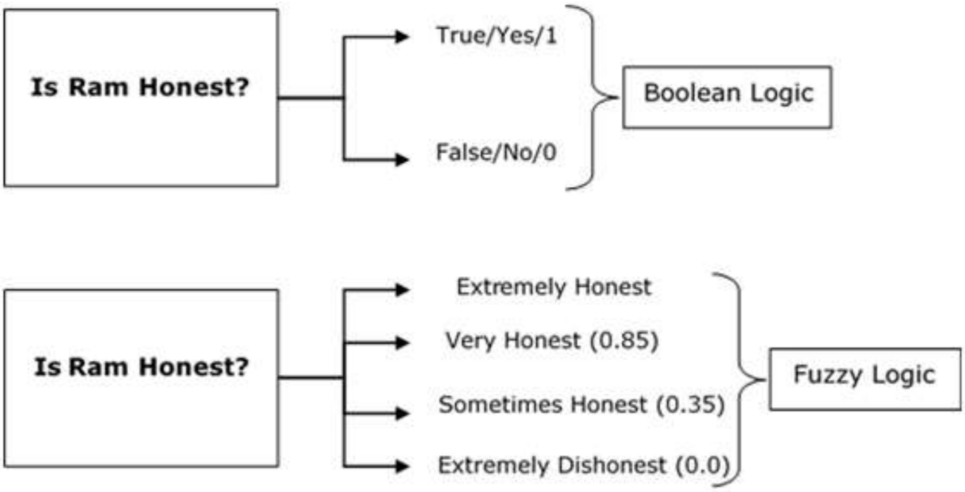
FUZZY SET THEORY

The word **fuzzy** refers to things which are not clear or are vague. Any event, process, or function that is changing continuously cannot always be defined as either true or false, which means that we need to define such activities in a Fuzzy manner.

What is Fuzzy Logic?

Fuzzy Logic resembles the human decision-making methodology. It deals with vague and imprecise information. This is gross oversimplification of the real-world problems and based on degrees of truth rather than usual true/false or 1/0 like Boolean logic.

Take a look at the following diagram. It shows that in fuzzy systems, the values are indicated by a number in the range from 0 to 1. Here 1.0 represents **absolute truth** and 0.0 represents **absolute falseness**. The number which indicates the value in fuzzy systems is called the **truth value**.



In other words, we can say that fuzzy logic is not logic that is fuzzy, but logic that is used to describe fuzziness. There can be numerous other examples like this with the help of which we can understand the concept of fuzzy logic.

Fuzzy Logic was introduced in 1965 by Lofti A. Zadeh in his research paper “Fuzzy Sets”. He is considered as the father of Fuzzy Logic.

Types of Sets

Sets can be classified into many types; some of which are finite, infinite, subset, universal, proper, singleton set, etc.

Finite Set

A set which contains a definite number of elements is called a finite set.

Example − $S = \{x | x \in N \text{ and } 70 > x > 50\}$

Infinite Set

A set which contains infinite number of elements is called an infinite set.

Example − $S = \{x | x \in N \text{ and } x > 10\}$

Subset

A set X is a subset of set Y (Written as $X \subseteq Y$) if every element of X is an element of set Y.

Example 1 − Let, $X = \{1,2,3,4,5,6\}$ and $Y = \{1,2\}$. Here set Y is a subset of set X as all the elements of set Y is in set X. Hence, we can write $Y \subseteq X$.

Example 2 − Let, $X = \{1,2,3\}$ and $Y = \{1,2,3\}$. Here set Y is a subset (not a proper subset) of set X as all the elements of set Y is in set X. Hence, we can write $Y \subseteq X$.

Proper Subset

The term "proper subset" can be defined as "subset of but not equal to". A Set X is a proper subset of set Y (Written as $X \subset Y$) if every element of X is an element of set Y and $|X| < |Y|$.

Example − Let, $X = \{1,2,3,4,5,6\}$ and $Y = \{1,2\}$. Here set $Y \subset X$, since all elements in Y are contained in X too and X has at least one element which is more than set Y.

Universal Set

It is a collection of all elements in a particular context or application. All the sets in that context or application are essentially subsets of this universal set. Universal sets are represented as U.

Example − We may define U as the set of all animals on earth. In this case, a set of all mammals is a subset of U, a set of all fishes is a subset of U, a set of all insects is a subset of U, and so on.

Empty Set or Null Set

An empty set contains no elements. It is denoted by Φ. As the number of elements in an empty set is finite, empty set is a finite set. The cardinality of empty set or null set is zero.

Example – $S = \{x | x \in N \text{ and } 7 < x < 8\} = \Phi$

Singleton Set or Unit Set

A Singleton set or Unit set contains only one element. A singleton set is denoted by $\{s\}$.

Example − $S = \{x | x \in N, 7 < x < 9\} = \{8\}$

Equal Set

If two sets contain the same elements, they are said to be equal.

Example − If $A = \{1,2,6\}$ and $B = \{6,1,2\}$, they are equal as every element of set A is an element of set B and every element of set B is an element of set A.

Equivalent Set

If the cardinalities of two sets are same, they are called equivalent sets.

Example − If $A = \{1,2,6\}$ and $B = \{16,17,22\}$, they are equivalent as cardinality of A is equal to the cardinality of B. i.e. $|A| = |B| = 3$

Overlapping Set

Two sets that have at least one common element are called overlapping sets. In case of overlapping sets −

$$n(A \cup B)=n(A)+n(B)-n(A \cap B)n(A \cup B)=n(A)+n(B)-n(A \cap B)$$

$$n(A \cup B)=n(A-B)+n(B-A)+n(A \cap B)n(A \cup B)=n(A-B)+n(B-A)+n(A \cap B)$$

$n(A)=n(A-B)+n(A\cap B)n(A)=n(A-B)+n(A\cap B)$

$n(B)=n(B-A)+n(A\cap B)n(B)=n(B-A)+n(A\cap B)$

Example − Let, A = {1,2,6} and B = {6,12,42}. There is a common element '6', hence these sets are overlapping sets.

Disjoint Set

Two sets A and B are called disjoint sets if they do not have even one element in common. Therefore, disjoint sets have the following properties −

(i) $n(A\cap B)=\phi n(A\cap B)=\phi$ (ii)$n(A\cup B)=n(A)+n(B)n(A\cup B)=n(A)+n(B)$

Example − Let, A = {1,2,6} and B = {7,9,14}, there is not a single common element, hence these sets are overlapping sets.

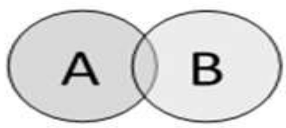
Operations on Classical Sets

Set Operations include Set Union, Set Intersection, Set Difference, Complement of Set, and Cartesian Product.

Union

The union of sets A and B (denoted by A∪ BA ∪ B) is the set of elements which are in A, in B, or in both A and B. Hence, A ∪ B = {x|x∈ A OR x ∈ B}.

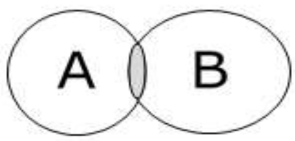
Example − If A = {10,11,12,13} and B = {13,14,15}, then A ∪ B = {10,11,12,13,14,15} – The common element occurs only once.



A ∪ B

Intersection

The intersection of sets A and B (denoted by A ∩ B) is the set of elements which are in both A and B. Hence, A ∩ B = {x|x∈ A AND x ∈ B}.

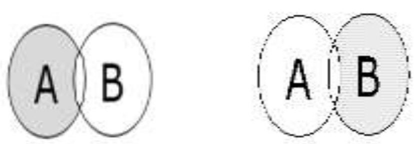


A ∩ B

Difference/ Relative Complement

The set difference of sets A and B (denoted by A–B) is the set of elements which are only in A but not in B. Hence, A − B = {x|x∈ A AND x ∉ B}.

Example − If A = {10,11,12,13} and B = {13,14,15}, then (A − B) = {10,11,12} and (B − A) = {14,15}. Here, we can see (A − B) ≠ (B − A)

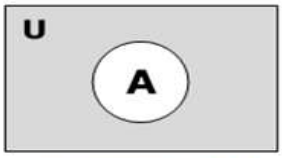


A-B B-A

Complement of a Set

The complement of a set A (denoted by A′) is the set of elements which are not in set A. Hence, $A' = \{x | x \notin A\}$.

More specifically, $A' = (U - A)$ where U is a universal set which contains all objects.



Complement of set A

Example − If A = {x|x belongs to

set of add integers} then A′ = {y|y does not belong to set of odd integers}

Properties of Crisp Sets

Properties on sets play an important role for obtaining the solution. Following are the different properties of classical sets −

Commutative Property

Having two sets **A** and **B**, this property states −

$$A \cup B = B \cup A A \cup B = B \cup A$$

$$A \cap B = B \cap A A \cap B = B \cap A$$

Associative Property

Having three sets **A**, **B** and **C**, this property states −

$$A \cup (B \cup C) = (A \cup B) \cup C A \cup (B \cup C) = (A \cup B) \cup C$$

$$A \cap (B \cap C) = (A \cap B) \cap C A \cap (B \cap C) = (A \cap B) \cap C$$

Distributive Property

Having three sets **A**, **B** and **C**, this property states −

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C) A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

Idempotency Property

For any set **A**, this property states −

$$A \cup A = A A \cup A = A$$

$$A \cap A = A A \cap A = A$$

Identity Property

For set **A** and universal set **X**, this property states −

$$A \cup \varphi = A A \cup \varphi = A$$

$$A \cap X = A A \cap X = A$$

$$A \cap \varphi = \varphi A \cap \varphi = \varphi$$

$$A \cup X = X A \cup X = X$$

Transitive Property

Having three sets **A**, **B** and **C**, the property states −

$$\text{If } A \subseteq B \subseteq C \subseteq B \subseteq C, \text{ then } A \subseteq C A \subseteq C$$

Involution Property

For any set **A**, this property states $-\overline{\overline{A}} = A\overline{\overline{A}} = A$

De Morgan's Law

It is a very important law and supports in proving tautologies and contradiction. This law states −

$$\overline{A \cap B} = \overline{A} \cup \overline{B} \overline{A \cap B} = \overline{A} \cup \overline{B}$$

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

Partition and Covering

Partition

A Partition on A is defined to be a set of non-empty subset Ai, each of which is pairwise disjoint and whose union yields the original set A.

Partition on A indicated as Π(A) is therefore

(i) $A_i \cap A_j = \phi$ for each pair(i,j) is element of I , $i \neq j$
(ii) Union $A_i = A$

Covering

A Coveringon A is defined to be a set of non- empty subset Ai, whose union yields the original set A. The non-empty subset need not

Operations on Fuzzy Sets

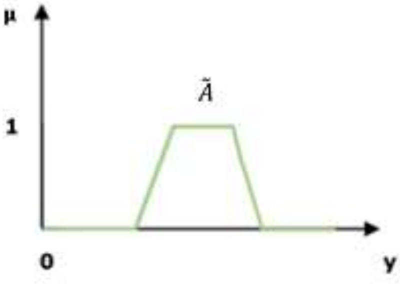
Having two fuzzy sets $\tilde{A}$A~ and $\tilde{B}$B~, the universe of information UU and an element y of the universe, the following relations express the union, intersection and complement operation on fuzzy sets.

Union/Fuzzy 'OR'

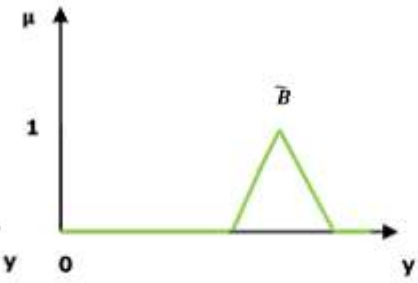
Let us consider the following representation to understand how the **Union/Fuzzy 'OR'** relation works −

$$\mu_{\tilde{A} \cup \tilde{B}}(y) = \mu_{\tilde{A}} \vee \mu_{\tilde{B}} \forall y \in U \mu_{A \sim \cup B \sim}(y) = \mu_{A \sim} \vee \mu_{B \sim} \forall y \in U$$

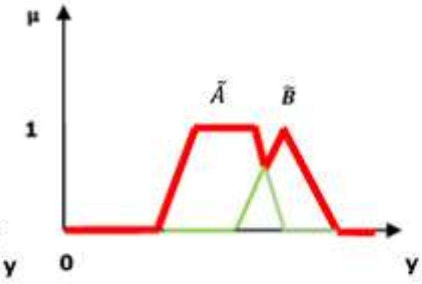
Here ∨ represents the 'max' operation.



Fuzzy set Ã



Fuzzy set B̃



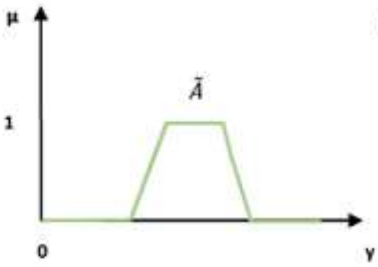
Union of two Fuzzy sets

Intersection/Fuzzy 'AND'

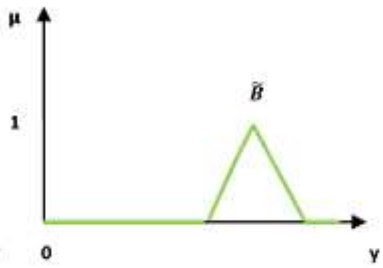
Let us consider the following representation to understand how the **Intersection/Fuzzy 'AND'** relation works –

μA˜∩B˜(y)=μA˜∧μB˜∀y∈UμA~∩B~(y)=μA~∧μB~∀y∈U

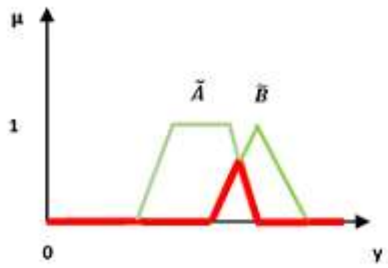
Here ∧ represents the 'min' operation.



Fuzzy set Ã



Fuzzy set B̃

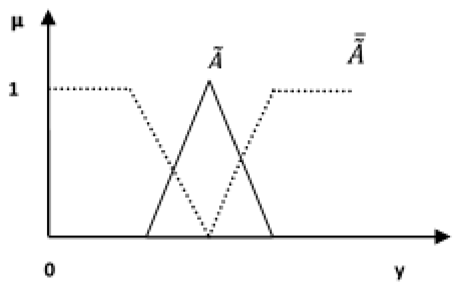


Intersection of two Fuzzy sets

Complement/Fuzzy 'NOT'

Let us consider the following representation to understand how the **Complement/Fuzzy 'NOT'** relation works –

μA˜=1−μA˜(y)y∈UμA~=1−μA~(y)y∈U



Complement of a fuzzy set

Cartesian Product

Let A1, A2...An be fuzzy sets in U1, U2 …Un, respectively. The Cartesian product of A1, A2... An is a fuzzy set in the space U1 x U2 x…x Un with the membership function as:

μA1 x A2 x…x An (x1, x2,…, xn) = min [μA1 (x1), μA2 (x2), …. μAn (xn)]

So, the Cartesian product of A1, A2, ….., An are donated by A1 x A2 x….. x An

Cartesian product: Example
Let A = {(3, 0.5), (5, 1), (7, 0.6)}
Let B = {(3, 1), (5, 0.6)} Find the product.
The product is all set of pairs from A and B with the minimum associated memberships
Ax B = {[(3, 3), min (0.5, 1)], [(5, 3), min(1, 1)], [(7, 3), min(0.6, 1)], [(3, 5), min(0.5, 0.6)], [(5, 5), min(1, 0.6)], [(7,5), min(0.6, 0.6)]}
= {[(3, 3), 0.5], [(5, 3), 1], [(7, 3), 0.6], [(3, 5), 0.5], [(5, 5),0.6], [(7, 5),0.6]}

Crisp Relations

The relation between any two sets is the Cartesian product of b the elements of A1 x A2 x….. x An

For X and Y universes X ×Y = {(x, y)|x Є X, y Є Y}

$$\mu_{x \times y}(x, y) = \begin{cases} 1, & (x, y) \in X \times Y \\ 0, & (x, y) \in X \times Y \end{cases}$$

This relation can be represented in a matrix format.

Fuzzy Relations

Fuzzy relations are mapping elements of one universe, to those of another universe, Y, through the Cartesian product of two universes. X, Universe X = {1, 2, 3}
R (X, Y)= {[(x, y), μR (x, y)]\ (x, y) Є (X × Y)}

Where the fuzzy relation R has membership function

μR (x, y) = μA X B (x, y) = min (μA (x), μB (y))

Since the fuzzy relation from X to Y is a fuzzy set in X ×Y,then the operations on fuzzy sets can be extended to fuzzy relations. Let R and S be fuzzy relations on the Cartesian space X × Y then:

Union:μR U S (x, y) = max [μR (x, y), μ S (x, y)]
Intersection: μR Π S (x, y) = min [μR (x, y), μ S (x, y)]
Complement: μR (x, y) = 1 - μR (x, y)

Fuzzy Relations: Example

Assume two Universes: A = {3, 4, 5} and B = {3, 4, 5, 6, 7}

$$\mu R(x, y) = \begin{cases} (y-x)/(y+x+2) & \text{if } y > x \\ 0, & \text{if } y \leq x \end{cases}$$

This can be expressed as follow:

$$\begin{array}{c} 3 \end{array} \left\{ \begin{array}{ccccc} 0 & 0.11 & 0.2 & 0.27 & 0.33 \end{array} \right.$$

R =	4	0	0	0.09	0.17	0.23
	5	0	0	0	0.08	0.14
		3	4	5	6	7

Fuzzy Relations: Example

This matrix represents the membership grades between elements in X and Y

μR (x, y) = {[0/(3, 3)], [0.11/(3, 4)], [0.2/(3, 5)],…………….., [0.14/(5, 7)]}

Assume two fuzzy sets: A = {0.2/x1+ 0.5/x2+ 1/x3}
B = {0.3/y1+ 0.9/y2}

Find the fuzzy relation (the Cartesian product)

$$A \times B = R = \begin{array}{c} x1 \\ x2 \\ x3 \\ \end{array} \begin{bmatrix} 0.2 & 0.2 \\ 0.3 & 0.5 \\ 0.3 & 0.9 \end{bmatrix}$$
$$\quad y1 \quad y2$$

Composition of Fuzzy Relations

Composition of fuzzy relations used to combine fuzzy relations on different product spaces. Having a fuzzy relation; R (X ×Y) and S (Y ×Z), then Composition is used to determine a relation T (X × Z),

Consider two fuzzy relation; R (X ×Y) and S (Y ×Z), then a relation T (X × Z), can be expressed as (max-min composition)

T = R o S μT (x, z) = max-min [μR (x, y), μS(y, z)] = V [μR (x, y) ^ μS (y, z)]

If algebraic product is adopted, then max-product composition is adopted:

T = R o S μT (x, z) = max [μR (x, y) .μS(y, z) = V [μR (x, y) . μS (y, z)]

Assume the following universes: X = {x1, x2}, Y = {y1, y2}, and Z = {z1, z2, z3}, with the following fuzzy relations.

$$R = \begin{array}{c} x1 \\ x2 \end{array} \begin{bmatrix} 0.7 & 0.5 \\ 0.8 & 0.4 \end{bmatrix} \text{and}$$
$$y1 \quad y2$$

$$S = \begin{array}{c} y1 \\ y2 \end{array} \begin{bmatrix} 0.9 & 0.6 & 0.2 \\ 0.1 & 0.7 & 0.5 \end{bmatrix}$$
$$z1 \quad z2 \quad z3$$

Find the fuzzy relation between X and Z using the max-min and max-product composition
By max-min composition μT (x1, z1) = max [min (0.7, 0.9), min (0.5, 0.1)] = 0.7

$$T = \begin{array}{c c} x1 & \\ x2 & \end{array} \begin{bmatrix} 0.7 & 0.6 & 0.5 \\ 0.8 & 0.6 & 0.4 \end{bmatrix}$$
$$\quad\quad z1 \quad z2 \quad z3$$

By max-product composition

μT (x2, z2) = max [(0.8, 0.6), (0.4, 0.7)] = 0.48

$$T = \begin{array}{c} x1 \\ x2 \end{array} \begin{bmatrix} 0.63 & 0.42 & 0.25 \\ 0.72 & 0.48 & 0.20 \end{bmatrix}$$
$$\quad\quad z1 \quad z2 \quad z3$$

FUZZY SYSTEMS

Logic

- Logic is the science of reasoning. Symbolic or mathematical logic has turned to be powerful computational paradigm.
- Just as mathematical sets have been classified into crisp sets and fuzzy sets.
- Logic can be broadly viewed as crisp logic and fuzzy logic.
- Crisp logic build on two states truth table (True/False).
- Fuzzy logic build on a multistage truth table.

Crisp Logic

- Consider the statements "Water boils at 90˚C" and "Sky is blue". An agreement of disagreement with these statements is indicated by True or False.
- Statement which is either true of false but not both is called Proposition.
- A Proposition is indicated by P,Q,R and so on.
- Example
  - P: Water boils at 90˚C.
  - Q: Sky is blue.

Sequence of propositions linked using connectors and operators.

- OR (V)
- AND (Λ)
- Negation/ NOT (¬)
- Implication / if-then (⇔)
- Equality (=)

.The truth table for the connectiveness

P	Q	PΛQ	PVQ	¬P	P⇔Q	P = Q
T	T	T	T	F	T	T
T	F	F	T	F	F	F
F	F	F	F	T	T	T
F	T	F	T	T	T	F

Laws of Propositional Logic

Propositional Logic supports following laws which can be effectively used for Simplification

(i) Commutativity - $(P\vee Q)=(Q\vee P)$
$(P\wedge Q)=(Q\wedge P)$
-
(ii) Associativity - $P\vee(Q\vee R)=(P\vee Q)\vee R$.
- $P\wedge(Q\wedge R)=(P\wedge Q)\wedge R$.

(iii) Distributivity - $P\vee(Q\wedge R)=(P\vee Q)\wedge(P\vee R)$.
- $P\wedge(Q\vee R)=(P\wedge Q)\vee(P\wedge R)$.

(iv) Identity - $P\vee F=P$
- $P\vee T=P$
- $P\wedge F= F$
$P\vee T = T$
-
(v) Negation - $P\vee\neg P=T$
- $P\wedge\neg P=F$

(vi) Idempotence - $P\vee P=P$
- $P\wedge P=p$

(vii) Absorption - $P\wedge((P\vee Q)=P$
$P\vee(P\wedge Q)=p$

(viii) *De Morgans's Law* - $\neg(P\vee Q)=\neg P\wedge\neg Q$.

$\neg(P\wedge Q)=\neg P\vee\neg Q$

(ix) *Involution* - $\neg(\neg P)=p$

Inference in Propositional Logic

- Inference is a technique by which, given a ser of facts and postulates or axioms or premises F_1, F_2, F_3, F_4,$F_n$goal G is to be derived.
- For example, from the facts, Where there is smoke there is fire". "There is smoke in the hill", the Statement "Then the hill is on fire" can be easily deduced.
- In propositional logic, three rules are widely used for inferring rules,
  - Modus ponens(mod pons)
  - Modus tollens
  - Chain rule

Modus ponens(mod pons)

Given P => Q and P to be true, Q is true,

p=> Q

P

$$\frac{\qquad}{Q}$$

Hence, the formulae above the line are the premises and the one below is the goal which can be inferred the premises.

Modus tollens

Given P => Q and ~Q to be true,~P is true,

$$\frac{\begin{array}{l} p \Rightarrow Q \\ \sim Q \end{array}}{\sim P}$$

Chain Rule

Given P => Q and Q=> R to be true, P =>Ris true,

$$\frac{\begin{array}{l} p \Rightarrow Q \\ Q \Rightarrow R \end{array}}{P \Rightarrow R}$$

Hence, the chain rule is a representation of the transitivity relation with respect to => connective.

Predicate Logic

- This logic deals with predicates, which are propositions containing variables.
- A predicate is an expression of one or more variables defined on some specific domain. A predicate with variables can be made a proposition by either assigning a value to the variable or by quantifying the variable.

Following are a few examples of predicates −

- Let E(x, y) denote "x = y"
- Let X(a, b, c) denote "a + b + c = 0"
- Let M(x, y) denote "x is married to y"

Predicate logic comprises the following apart from connectives and proposition recognized by propositional logic.

- Constants
- Variables
- Predicates
- Quantifiers
- Functions

Constants

Constant represents objects that do not change values.

Ex: Pencil, Ram

Variables

Variables are symbols which represent values acquired by the objects qualified by the quantifier with which they are associated with.

Ex: x,y,z

Predicates

Predicates are representative of association between constant and variables. A predicate carries an name representing the association followed by arguments representing the objects it is to associate.

Ex: likes(Ram,tea) (Ram likes tea)

plays(sita,x) (sita plays anything)

Quantifiers

Quantifiers are symbols which indicates the two types of quantification namely, All(∀.) and some(∃.).∀ is termed as Universal quantifier and ∃ is termed as Existential quantifier

Ex:

All men are mortal

Some mushrooms are poisonous.

∀x(man(x) => mortal(x)

∃x mushroom(x) => poisonous(x)

Functions

Function are similar to predicates in form and their representation of association between objects but unlike predicates which acquire truth values alone, functions acquire values other thantruth values

Ex: plus(2,3) (2 plus 3 which is 5)

Mother(krishna) (Krishna's mother)

Interpretations of Predicate Logic Formula

- For a formula in propositional logic, depending on the truth values acquired by the propositions, the truth table interprets the formula.
- But in the case of predicate logic, depending on the truth values acquired by the predicates, the nature of quantifiers, and the values taken by the constants and functions over a domain D, the formula is interpreted.

Inference in Predicate Logic

- The rules of inference such as Modus ponens, Modus tollens, and chain rule, and the laws of propositional logic are applicable for inferring predicate logic but not before the quantifiers have been appropriately eliminated.

Fuzzy Logic

- In Crisp logic, the truth values acquired by propositions or predicates are 2-values, namely True, False which may be treated numerically equivalent to (0,1).
- In Fuzzy logic truth values are multi-valued such as absolutely true, partly true, absolutely false, very true and so on and are numerically equivalent to (0-1).

Fuzzy Proposition

- A fuzzy proposition is a statement which acquires a fuzzy truth table.
- Fuzzy propositions are associated with fuzzy sets.
- Fuzzy membership value associated with the fuzzy set A' for p' is treated as fuzzy truth table T(P')F

Fuzzy Connective

Fuzzy logic similar to crisp logic supports the following connectives:

(i) Negation : ¬

(ii) Disjunction : ∧

(iii) Conjunction : ∨

(iv) Implication : =>

Fuzzy Quantifiers

- Just as in crisp logic where predicates are quantified by quantifiers, fuzzy logic propositions are quantified by fuzzy quantifiers.
- There are two classes of fuzzy quantifiers such as
  - Absolute quantifiers and
  - Relative quantifiers

  While absolute quantifiers are defined over R̂, relative quantifiers are defined over {0-1].

Example

Absolute Quantifier	Relative Quantifier
round about 250	almost
Much greater than 6	about
some where around 20	most

Fuzzy Inference

- Fuzzy inference referred to as approximate reasoning refers to computational procedure used for evaluating linguistic descriptions.
- The two important inferring procedures are
  - Generalized Modulus Ponens(GMP)
  - Generalized Modulus Tollens(GMT)

GMP is formally stated as

IF x is Ă THEN y is $\bar{B}$

$\underline{\text{x is Ă'}}$

y is B'

Fuzzy rule-based systems

- *Fuzzy rules* are linguistic IF-THEN- constructions that have the general form "IF A THEN B" where A and B are (collections of) propositions containing linguistic variables.
- A is called the *premise* and B is the *consequence* of the rule. In effect, the use of linguistic variables and fuzzy IF-THEN- rules exploits the tolerance for imprecision and uncertainty. I
- n this respect, fuzzy logic mimics the crucial ability of the human mind to summarize data and focus on decision-relevant information.

In a more explicit form, if there are *I* rules each with *K* premises in a system, the i^{th} rule has the following form.

$$If \quad a_1 \; is \; A_{i,1} \; \Theta \; a_2 \; is \; A_{i,2} \; \Theta \ldots \Theta \; a_k \; is \; A_{i,k} \quad then \quad B_i$$

In the above equation *a* represents the crisp inputs to the rule and *A* and *B* are linguistic variables. The operator 1 can be AND or ORor XOR.

Example:

If a HIGH flood is expected and the reservoir level is MEDIUM, then water release is HIGH.

Defuzzification

- It is the process of producing a quantifiable result in crisp logic iven fuzzy sets and corresponding membership degrees.
- It is the process that maps a fuzzy set to a crisp set. It is typically needed in fuzzy control systems.
- These will have a number of rules that transform a number of variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets.
- For example, rules designed to decide how much pressure to apply might result in "Decrease Pressure (15%), Maintain Pressure (34%), Increase Pressure (72%)".
- Defuzzification is interpreting the membership degrees of the fuzzy sets into a specific decision or real value.
- The simplest but least useful defuzzification method is to choose the set with the highest membership, in this case, "Increase Pressure" since it has a 72% membership, and ignore the others, and convert this 72% to some number.
- The defuzzifcation using
  - Centriod method.
  - Centre of sums method.
  - Mean of Maxima method.

Applications

Two applications of fuzzy control system are

- Fuzzy Crusie Control System
- Air conditioner Controller

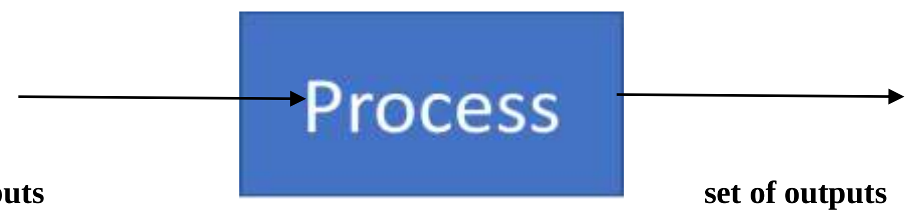
UNIT-V
Genetic Algorithms

Introduction

- Genetic Algorithm(GA) is search-based optimization techniques based on the principles of Natural Genetics and Natural selection.
- It finds the Optimal Solution.
- It is based on the survival of the Fittest.

Optimization

- Optimization is the process of making something better.
- Optimization refers to finding the values of inputs in such a way that we get the “best” output values.
- The definition of “best “ varies from problem to problem
- In mathematical terms, it refers to maximizing and minimizing one of more objective functions, by varying the input parameters.
-



set of inputs **set of outputs**

Search Space

- The set of possible solutions.
- In this search space, lies a point or set of points which gives the optimal solution.
- The aim of the optimization is to find that point or set of points in the search space.

Nature Genetics

- Solutions then undergo Crossover and mutation.
- Producing new children
- The process is repeated over various generations.

Genetic Algorithm

- Decision making features occur in all fields of human activities such as scientific and technological and affect every sphere out life.

- Engineering design, which entails sizing, dimensioning and detailed element planning.
- The aim is to make objective function a maximum or minimum, that is, it is required to find element Xo in A if it exists such that

  $F(X_o) < F(X)$ for minimization

  $F(X) < F(X_o)$ for maximization

Several optimization problems have not been tackled by classical procedure including

- Linear programming
- Transportation
- Assignment
- Nonlinear programming
- Dynamic programming
- Inventory
- Queuing
- Replacement
- Scheduling

Classification of Optimization Techniques

The search is of two kinds namely

- Deterministic
- Non-deterministic

In deterministic search, algorithm methods such as Steepest Gradient methods are employed.

- In stochastic approach, random variables are introduced.
- Whether the search is deterministic or stochastic, it is possible to improve the reliability of the result.
- A transition rule may be used to improve reliability.

Classification of Searching Techniques

Non- traditional search and optimization methods become popular in engineering optimization problem. These algorithm include:

- Simulated annealing – cooling phenomenon of molten metals to constitute search procedure.
- Ant colony Optimization – The collective behavior emerges from group of social insects such as ants, bees, swaps and termites has been dubbed as swarm intelligence.
- Random cost – Stochastic algorithm which moves enthusiastically uphill or downhill.
- Evolution Strategy -

- Genetic Algorithms - both use natural genetics and natural selection to construct search and optimization procedure.
- Cellular Automata – based on automata theory

Genetic Algorithms – History

- The idea of evolutionary computing as introduced in 1960 by I.Rechenburg in his work Evolutionary Strategies.
- Genetic Algorithms are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection.
- Prof. Holland fo University of Michigan, Ann Arbor, envisaged the concept of GA in the mid-sixties and published his seminar work in 1975.
- Many fields in which GAs have been successfully includes biology, computer science, image processing and pattern recognition, physical science, social sciences and neural networks.

Basic Concepts

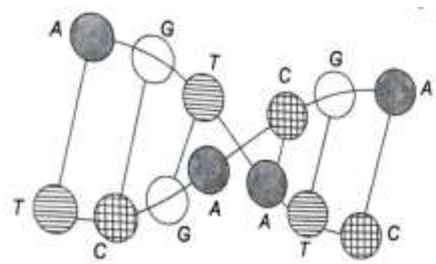
- *Genetic algorithms are good at taking larger, potentially huge, search spaces and navigating them looking for optimal combinations of things and solutions which we might not find in a life time.*

- GA are different from traditional optimization methods.
- Gas need design space to be converted into genetic space.
- A more striking difference between GA and traditional optimization methods is that GA uses population of points at one time in contrast to single point approach by traditional method.
-
- In traditional method, transition rules are used and they are deterministic in nature, but GA used randomized operator.
- Random operators improve the search space in an adaptive manner.
- Three most important aspects of using GA are:
  - Definition of objective function.
  - Definition and implementation of genetic representation.
  - Definition and implementation of genetic operator.

Biological Background

- All living organisms consists of cells. In each cell, there is a set of chromosomes which are strings of DNA and serve as a model for the whole operation.
- A chromosome consists of genes on blocks of DNA. Each gene encodes generate particular pattern.
- Each gene encodes a trait e.g. color of the eyes.
- Possible settings of traits are called **alleles**.

- Each gene has it own position in the chromosome work space. This position is called locus.
- Complete set of genetic material is called genome and a particular set of genes in genome is called genotype.



Genome consisting of Chromosomes

Creation of Offsprings

- During the creation of offspring, recombination occurs(crossover) and in that process genes from parents form a whole new chromosome in some way.
- The new created chromosome can be mutated.
- Mutated means elements of DNA can be modified.
- These changes are mainly caused by errors in copying genes from parents.
- The fitness of an organism is measured by means of success of organism in life.

Search Space

- The space for all possible feasible solutions is called search space.
- Each solution can be marked by its value of the fitness of problem.
- "Looking for a solution" means looking for extreme (maximum or minimum) in search space.
- GAs inspired by Darwinian theory of the survival of the fittest.
- Algorithm started with set of new solutions called population.
- Solution for one population are taken and used to from new population.
- Solution, which are selected to from new population are offspring are selected according their fitness.
- This is repeated until some condition for improvement of best solution is satisfied.

Working Principle

First we consider unconstrained optimization problem

$$\text{maximize } f(X)$$
$$X_i^L \leq X_i \leq X_i^U \text{ for } i=1,2,\ldots N$$

If we want to minimize f(X), for f(X)>0, the we can write the objective function as maximize 1/1+f(X)

IF f(X)<0 instead of minimizing f(X), maximize {-f(X)}. Hence, both maximization and minimizaion problems can be handled by GA.

If the same problem is solved by multiple regression analysis, given K dependency variable 2(K+1)-1 including the intercept which are given

Subset of Regression Analysis

Variable	Subsets
2	7
3	15
-	=
-	=
9	1023
19	10,48,578

On the other hand, in GA the variables are coded.

Encoding

- There are many ways of representing individual genes.
- Holland worked mainly with string bits but we can use arrays, trees, lists or any other object.
- Here we consider only bit strings.

Binary Encoding

- Binary Encoding gives many possible chromosomes even with small number of alleles.
- On the other hand, encoding is often not natural for many problems and sometimes corrections must be made after genetic operator corrections.

Chromosome A	101101100011
Chromosome B	010011001100

Chromosomes

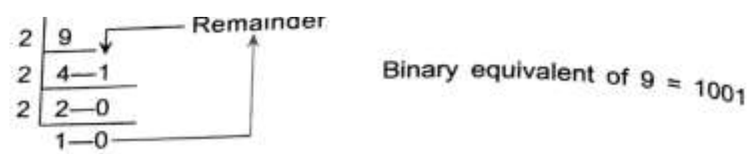
- The length of the string usually determined according to the desired solution accuracy.

For example, 4-bit binary string can be used to represent 16 numbers

4-bit String	Numeric Value	4-bit String	Numeric Value	4-bit String	Numeric Value
0000	0	0110	6	1100	12
0001	1	0111	7	1101	13
0010	2	1000	8	1110	14
0011	3	1001	9	1111	15
0100	4	1010	10		
0101	5	1011	11		

4-Bit String

To convert any integer to binary string , go on dividing the integer by 2



we represent to variables X_1, X_2 as (1011 0110). Every variable will have both upper and lower limit as

$$X_i^L \leq X_i \leq X_i^U$$

n-bit string can represent integers from 0 to $2^n - 1$. Assume that X; is coded as a substring S_i of length n_i. The decoded value of a binary substring is calculated as

$$\sum_{k=0}^{k=n_i-1} 2^k s_k$$

where si can be either zero or 1 and the string S is represented as

$$S_{n-1} \ldots s_3 s_2 s_1 s_0$$

For example, a four-bit string (0111) has a decoded value equal to

$$2^3 \text{ x } 0 + 2^2 \text{ x } 1 + 2^1 \text{ x } 1 + 2^0 \text{ x } = 7$$

Knowing X_i^L and X_i^U corresponding to (000) and (1111), the equivalent value for any 4-bit string can be obtained as

$$X_i = X_i^L + \frac{(X_i^U - X_i^L)}{(2^{n_i} - 1)} \times \text{(decoded value of string)}$$

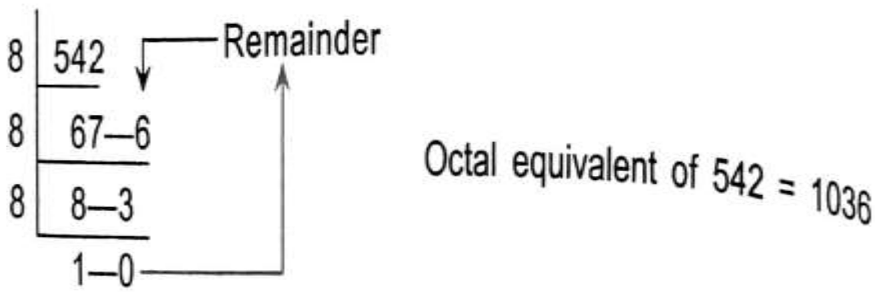
for example

$$S_i = 1010 = 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 1 + 2^0 \times 0 = 10$$

$$X_i = 2 + \frac{(17 - 2)}{(2^4 - 1)} \times 10 = 12$$

Octal Encoding

- To convert any integer to an octal string, go dividing the integer by 8 as shown below



Octal Encoding

For the octal code, we can get the equivalent integer by decoding it as shown, the integer code for 1036 is 542

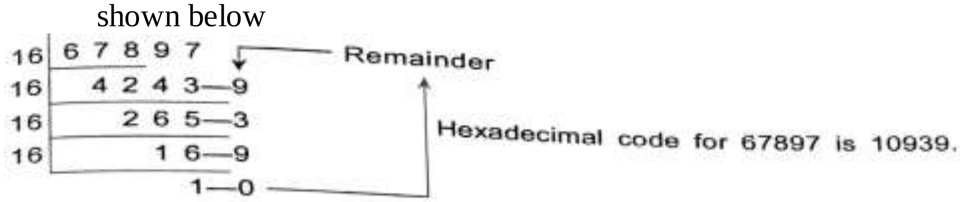
A four bit octal string represent integer from 0 to 4095 and hence, (0000, 0000) and (7777,7777) would represent the points for X_1 and X_2 as $(X_1^{L,}$ $,X_2^{L})$, (X_1^{L},X_2^{L}) ,

The decoded value of binary substring S_i is calculated as

$$\sum_{k=0}^{k=n_i-1} 8^k s_k$$

Hexadecimal Encoding

- To convert any integer to an hexadecimal string, go dividing the integer by16 as shown below



A four bit hexadecimal string represent integer from 0 to 65535 and hence, (0000, 0000) and (1111,1111) would represent the points for X_1 and X_2 as $(X_1^{L,}$ $,X_2^{L})$, (X_1^{L},X_2^{L}) ,

The decoded value of binary substring S_i is calculated as

$$\sum_{k=0}^{k=n_i-1} 16^k s_k$$

Encoding can be given to any base b, bits of n_i length can represent integer from 0 to $(b^{ni}$

-1) and hence (0000,0000) and (b-1) (b-1)

The decoded value of binary substring S_i is calculated as

$$\sum_{k=0}^{k=n_i-1} b^k s_k$$

Permutation Encoding

- This can be used in ordering problem such as travelling salesman or task ordering.
- Every chromosome is a string of numbers which represents the number in the sequence as shown

Chromosome - A	1	5	3	2	4	7	9	8	6
Chromosome - B	8	5	6	7	2	3	1	4	9

Permutation Encoding

Value Encoding

- Every chromosome is a string of some vales and the values can be any thing connected to the problem.
- From numbers, real numbers characterize some complicated objects as shown.

Chromosome - A	1.234	5.3243	0.4556	2.025s
Chromosome – B	abdjetidhj			
Chromosome – C	(Back)	(Right)	(Forward)	(Left)

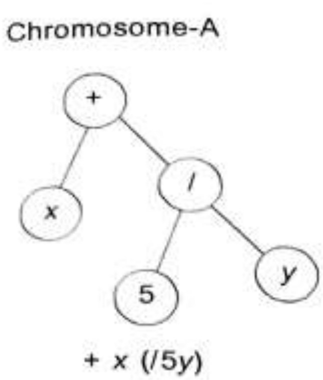
Value Encoding

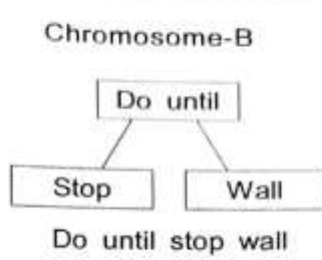
- Value encoding is very good for some special problems. On the other hand, this encoding is often necessary to develop new genetic operators specific to the problem.

Tree Encoding

- Every chromosome is a tree of some objects such as functions and commands,

Tree encoding is good for evolving programs ina programming language.





G.Umarani

Tree Encoding

Fitness Function

- GAs mimic the Darwinian Theory of survival of the fittest and principle of nature to make search process.
- GAs are usually suitable for the maximization problems.
- Minimization problems are usually transformed into maximization problem by suitable transformation.
- In general Fitness function F(X) is derived from the objection function and used in successive genetic operations.
- Certain genetic operators require that fitness function be non-negative, although certain operators don't have these requirements.
- Consider the following transformation,

$F(X)=f(X)$ for maximization problem

$F(X)=1/f(X)$ for minimization problem, if $f(X)\neq 0$

$F(X)=1/(1+f(X))$, if $f(X)=0$

A number of such transformations are possible. The fitness function value of the string is known as string's fitness.

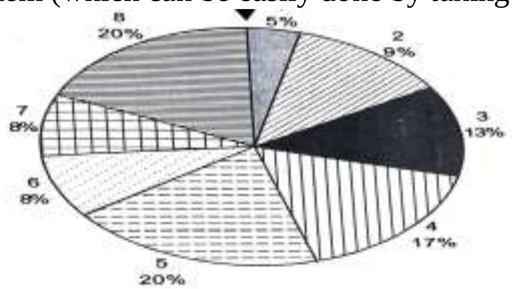
Reproduction

- Reproduction is usually the first operator applied on population.
- Chromosomes a selected from the population to be patents to cross over and produce offspring.
- According to the survival of the fittest, the best ones should survive and create new offspring.
- Reproduction operator is sometimes known as selection operator.
- The various methods of selecting chromosomes for parents to crossover are:
    - Roulette Wheel Selection
    - Boltzmann Selection
    - Tournament Selection
    - Rank Selection
    - Steady-state Selection

Roulette Wheel Selection

- In the roulette wheel selection, the probability of choosing an individual for breeding of the next generation is proportional to its fitness, the better the fitness, the higher chance for that individual to be chosen.
- Choosing individuals can be depicted as spinning a roulette that has as many pockets as there are individuals in the current generation, with sizes depending on their probability.
- Probability of choosing individual is equal to, where is the fitness and is the size of

  current generation.
- If we're working on minimization problem, it is however needed to transform it into maximization problem (which can be easily done by taking the inversion of our fitness).



Roulette Wheel marked for each individual according to selection

The average fitness is

$$\overline{F} = \sum_{j=1}^{n} F_j / n$$

Boltzmann Selection

- In Boltzmann selection, a continuously varying temperature controls the rate of selection according to a preset schedule.
- The temperature starts out high, which means that the selection pressure is low.
- The temperature is gradually lowered, which gradually increases the selection pressure, thereby allowing the GA to narrow in more closely to the best part of the search space while maintaining the appropriate degree of diversity.
- In thermal equilibrium, at a temperature T has it energy distributed probabilistically according to

$$P(E) = \exp(-E/KT)$$

where k is Boltzmann constant.

Tournament Selection

- GA uses a strategy to select the individuals from population and insert them into mating pool.
- Individuals from the mating pool are used go generate new offspring, which are the basis for next generation.
- The best individual (the Winner) from the tournament is the one with highest fitness.
- Tournament competitor and the winner are inserted into the mating pool,
-

Individuals	1	2	3	4	5	6	7	8
Fitness	1	2.10	3.11	4.01	4.66	1.91	1.93	4.55

Fitness values of Individuals

Step 1: First select individuals 2 and 4 at random

4 is the winner based on high fitness value

Step 2: Select individuals 3 and 8 at random

8 is the winner

Step 3: nest select 1 and 3

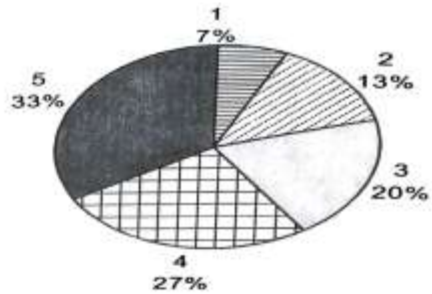
3 is the winner

Individuals	Selected
4 and 5	5
1 and 6	6
1 and 2	2
4 and 2	4
8 and 3	3

So the tournament Selection selected 4 and 8 as the mating pools.

Rank Selection

- Rank selection first ranks the population and taken every chromosomes, receives fitness from the ranking. The Roulette wheel selection is applied to the modified wheel as



Roulette wheel according to Rank.

Elitism

- Elitism can very rapidly increase the performance of GA because it prevents loosing the best-found solutions
- Goldberg suggest that the fitness of any i^{th} individual must be subtracted from the large constant.
- The new expression for fitness becomes

$$\O_i=(F_{max}-F_{min})=F_i(X)$$

- The process of reproduction confirms the Darwinian principles of survival of the fittest.

Generation Gap and Steady-state Replacement

- The generation gap is defined as the proportion of individuals in the population, which are replaced each generation.
- Population is replaced in each generation,
- Whitley proposed steady state replacement operates at the other extreme and in each generation only by few are replaced.
- Generation Gap can be classifies as

$$G_P=P/N_P$$

where N_p is the population size and p is the number of individuals that are replaced. Some of which are

- Selection of parents according to fitness and selection of replacement at random.
- Selection of parents at random and selection of replacement by inverse fitness.
- Selection of both parents and replacements according to fitness/inverse fitness.

Generation gap can be gradually increased as the evolution takes place to widen exploration space and may lead to better results.

Genetic Modelling

Generic Algorithm (GA) are search procedure based on mechanics of natural genetics and natural selection. The GA derives power from the genetic operators. Low level operators, namely

- Inversion
- Dominance
- Deletion
- Intrachromosomal duplication
- Translocation
- Segregation
- Speciation
- Migration
- Sharing
- Mating

A simple GA largely uses three basic operators

1. Reproduction
2. Cross Over
3. Mutation

CROSS OVER

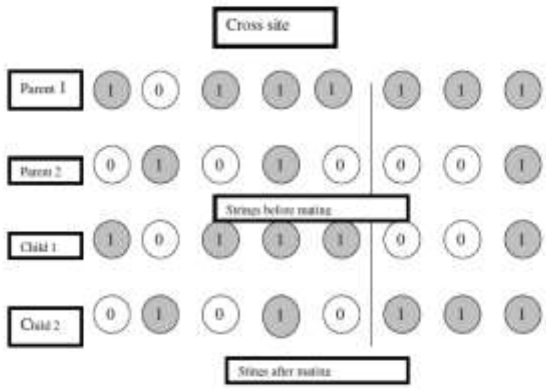
After the reproduction phase is over, the population is enriched with better individuals. Reproduction makes clones of good strings, but does not create new ones. Cross over operator is applied to the mating pool with a hope that it would create a better string. The aim of the cross over operator is n to search the parameter space. In addition, search is to be made in a way that n the information stored in the present string is maximally preserved because these parent strings are instances of good strings selected during reproduction.

Cross over is a recombination operator, which proceeds in three steps. First, the reproduction operator selects at random a pair of two individual strings for mating, then a cross-site is selected at random along the

string length and the position values are swapped between two strings following the cross site. For instance, let the two selected strings in a mating 95 pair be A = 11111 and B= 00000. If the random selection of a cross-site is two, then the new strings following cross over would be A* = 11000 and B* = 00111. This is a single–site cross over. Though these operators look very simple, their combined action is responsible for much of GA’s power. From a computer implementation point of view, they involve only random number of generations, string copying and partial string swapping. There exist **many types of cross over** operations in genetic algorithm

Single Site Cross Over

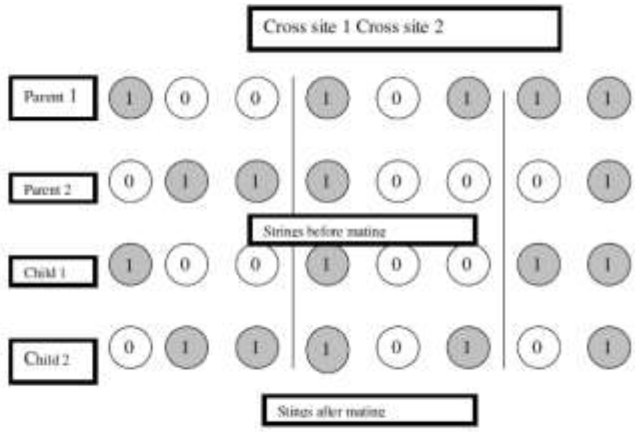
In a single – site cross over, a cross-site is selected randomly along the length of the mated strings and bits next to the cross-sites are exchanged.



If an appropriate site is chosen, better children can be obtained by combining good substances of parents. Since the knowledge of the appropriate site is not known and it is selected randomly, this random selection of cross-sites may produce enhanced children if the selected site is appropriate. If not, it may severely hamper the string quality. Anyway, because of the crossing of parents better children are produced and that will continue in the next generation because reproduction will not select those strings for the next mating pool.

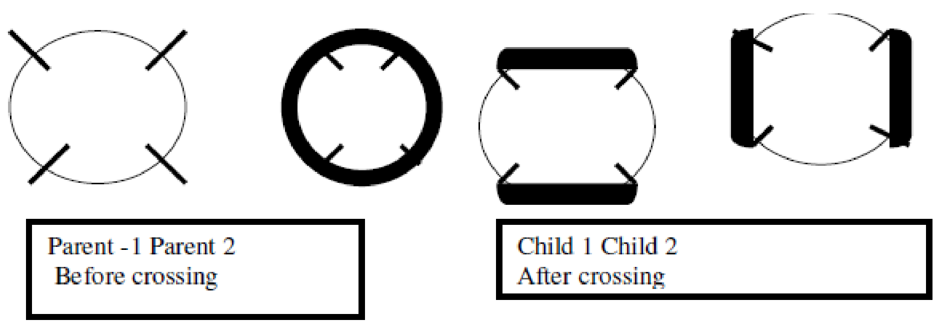
Two Point Cross Over

In a two – points cross over operator, two random sites are chosen and the contents bracketed by these sites are exchanged between two mated parents. If the cross-site 1 is three and cross-site 2 is six, the strings between three and six are exchanged.



Multi Point Cross Over

In a multi – point cross over, again there are two cases. One is theeven number of cross-sites and second one is the odd number of cross- sites.In case of even numbered cross-sites, the string is treated as a ring with nobeginning or end. The cross-sites are selected at random around the circleuniformly. Now the information between alternate pairs of sites in interchanged.



Uniform Cross Over

An extreme of multi-point cross over is the uniform cross over operator. In a uniform cross over operator, each bit from either parent is selected with a probability of 0.5 and then interchanged as shown in Figure (a). It is seen that uniform cross over is radically from one-point cross over. Sometimes gene in the offspring is created by copying the corresponding gene from one or the other parent chosen according to a randomly generated cross over mask. When there is 1 in the mask, the gene is copied from the first parent and when there is 0, the gene is copied from second parent as shown in fig (b). The process is repeated with the parents exchanged to produce the second offspring. A new cross over mask is 98 randomly generated for each pair of parents. Offspring therefore contains a mixture of genes from each parent. The number of effective crossing points is not fixed but average to L/2 (where L is chromosome length). Fig (a)

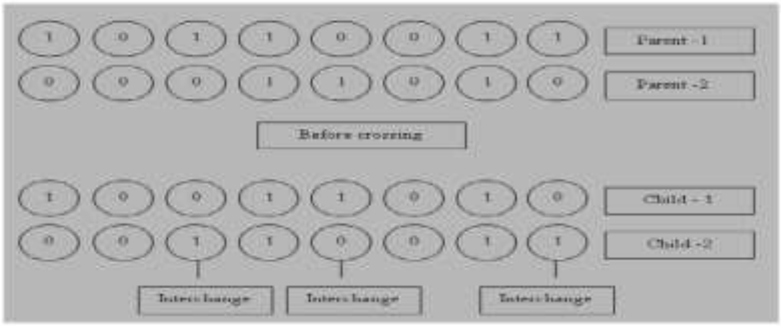
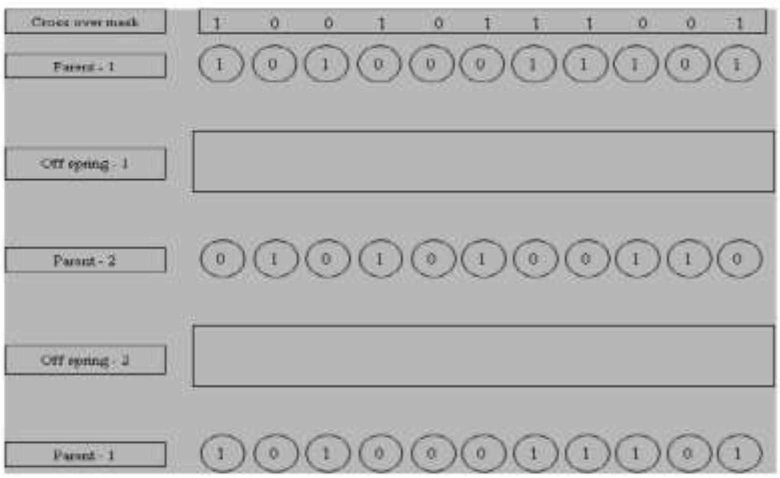


Figure b:



Cross Over Rate

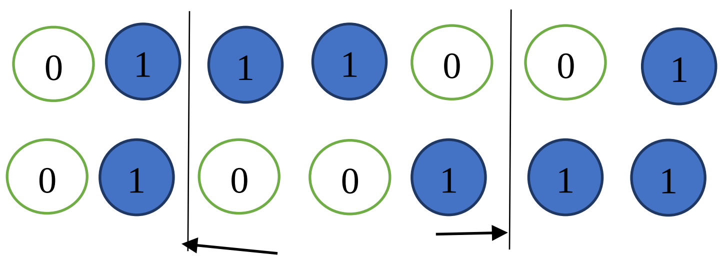
In GA literature, the term cross over rate is usually denoted as Pc.The probability varies from 0 to 1. This is calculated in GA by finding out the ratio of the number of pairs to be crossed to some fixed population. Typically for a population size of 30 to 200, cross over rates are ranged from 0.5 to 1. It is seen that with random cross-sites, the children strings produced may not have a combination of good substrings from parent strings depending on whether or not the crossing site falls in the appropriate place. But we do not worry about this too much because if good strings are created by cross over, there will be more copies of them in the next mating pool generated by the reproduction operator. But if good strings are not created by cross over, they will not survive too long, because reproduction will select against those strings in subsequent generations. It is clear from this discussion that the effect of cross over may either be detrimental or beneficial.

Thus, in order to preserve some of good strings that are already present in the mating pool, not all strings in the mating pool are used in cross over. When a cross over probability of Pc is used only 100 Pc percent strings in the population are used in the cross over operation and 100 (1- Pc) percentage of the population remains as it is in the current population. Even though the best 100 (1- Pc) % of the current population can be copied deterministically to the new population, this is usually preferred at random. A cross over operation is mainly responsible for the search of new strings.

INVERSION AND DELETION :

Inversion :

A String from the population is selected and the bits between two random sites are invertd.



Bits between sites inverted

Linear + end inversion :

Linear + end –inversion performs linear inversion with a specified probability of 0.75. If the linear inversion was not performed, the end inversion would be performed with equal probability of 0.0125 at either the left or right end of the string.

Continuous inversion:

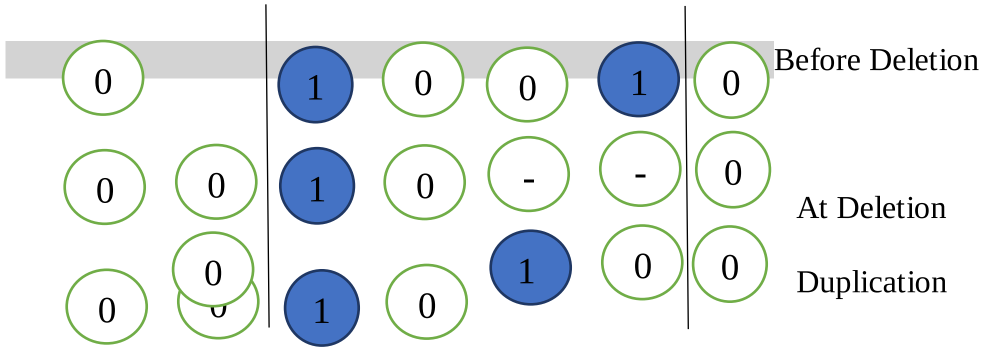
In *Continuous inversion*, inversion was applied with specified inversion probability P_r to each new individual when it is created.

Mass inversion:

No inversion takes place until a new population is created and thereafter, one-half of the population undergoes identical inversion.

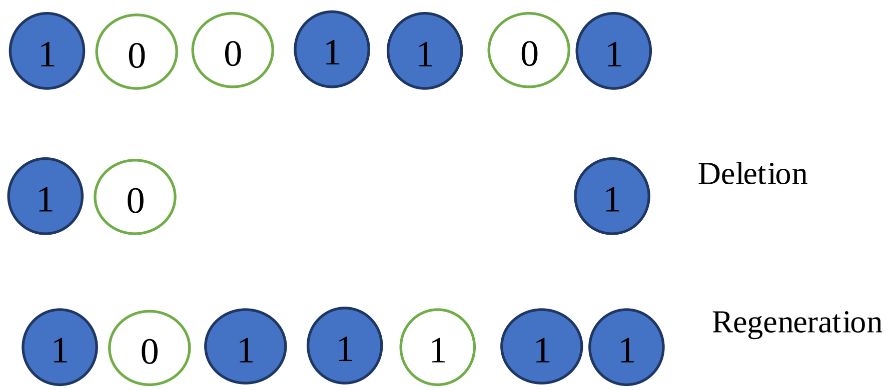
Deletion and Duplication:

Any two are three bits at random in order are selected and the previous bits are duplicated and it shown fig



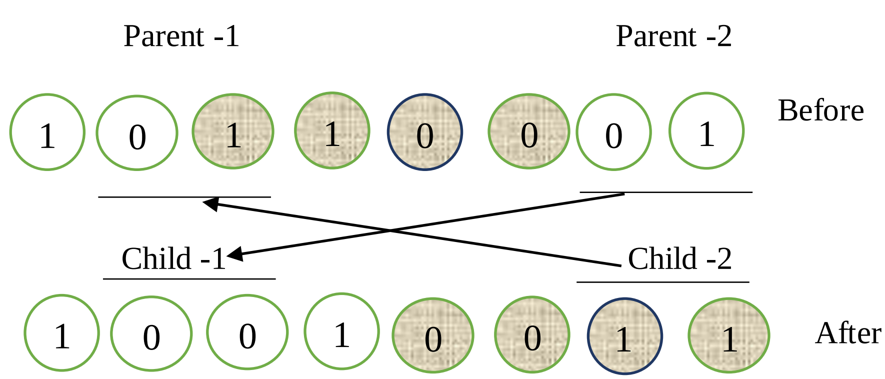
Deletion and Regeneration:

Genes between two cross –sites are deleted and regenerated randomly.



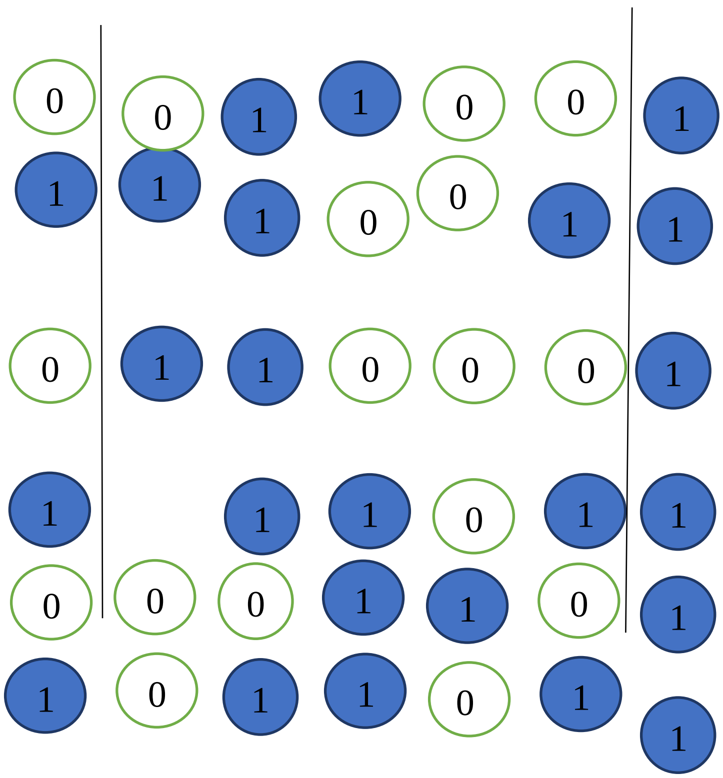
Segregation:

The bits of the parents are segregated and then crossed over to produce offspring.



Cross Over and Inversion:

Cross Over and inversion operator is the combination of both cross over and inversion operators. In this ,two random sites are chosen, the contents bracketed by these sites are exchanged between two mated parents and, the end points of these exchanged contents switch place.



Parent -1

Before cross over and inversion

Parent - 2

After Cross over

After Cross over and inversion

MUTATION OPERATOR

Mutation

After cross over, the strings are subjected to mutation. Mutation of a bit involves flipping it, changing 0 to 1 and vice versa with a small mutation probability Pm. The bit-wise mutation is performed bit-by-bit by flipping acoin with a probability of Pm. Flipping a coin with a probability of Pm is simulated as follows. A number between 0 to 1 is chosen at random. If the random number is smaller than Pm then the outcome or coin flipping is true, otherwise the outcome is false. If at any bit, the outcome is true then the bit is altered, otherwise the bit is kept unchanged. The bits of the strings are independently muted, that is, the mutation of a bit does not affect the probability of mutation of other bits.

A simple genetic algorithm treats the mutation only as a secondary operator with the role of restoring lost genetic materials. Suppose, for example, all the strings in a population have conveyed to zero at a given position and the optimal solution has one at that position and then cross over cannot regenerate one at that position while a mutation could. The mutation is simply as insurance policy against the irreversible loss of genetic material. The mutation operator introduces new genetic structures in the population by randomly modifying some of its building blocks. It helps the search algorithm to escape from local minima's traps since the modification is not related to any previous genetic structure of the population.

For example

0110 1011
0011 1101
0001 0110
0111 1100

It creates different structure representing other sections of the search space. The mutation is also used to maintain introduces some probability (Npm) of turning zero to one. Hence, mutation causes movement in the search space (local or global) and restores lost information to the population. 101

0110 1011

0011 1101

0001 0110

1111 1100

Mutation Rate Pm

Mutation rate is the probability of mutation which is used to calculate number of bits to be muted. The mutation operator preserves the diversity among the population which is also very important for the search. Mutation probabilities are smaller in natural populations leading us to conclude that mutation is appropriately considered a secondary mechanism of genetic algorithm adoption. Typically, the simple genetic algorithm uses the population size of 30 to 200 with the mutation rates varying from 0.001 to 0.5.

BIT – WISE OPERATOR

Binary coding is used more extensively in the coding mechanism to generate algorithm structure.

One's Complement operator

The One's complement operator(~) is an unary operator that causes the bits of its operand to be inverted, so that 1 becomes zero and zero becomes one.

A= 0100 0001 -> 4 1

B= 1011 1110 -> 11 14

Logical Bit –wise Operator

There are three Bit –wise operator 1.Bit wise AND (&), 2.exclusive OR (^), 3. Bit wise OR (!)

Bit wise AND (&) operator

A bit –wise AND (&) expression returns 1 if both the bits have a value 1, otherwise it returns a value 0.

Parent – 1a = 1010 1010 -> 10 10

Parent – 2b = 1100 0011 -> 12 03

Child a & b = 1000 0010 - > 8 2

Bit wise exclusive OR (^) operator

A bit –wise exclusive OR (^) expression returns 1 if one of the bits have a value 1 and the other has a value of 0, otherwise it returns a value 0.

Parent – 1a = 1010 1010 -> 10 10

Parent – 2b = 1100 0011 -> 12 03

Child a & b = 0110 1001 - > 6 9

Bit wise OR (I) operator:

A bit –wise OR (I) expression returns 1 if one or more bits have a value 1, otherwise it returns a value 0.

Parent – 1a = 1010 1010 -> 10 10

Parent – 2b = 1100 0011 -> 12 03

Child a & b = 1110 1011 - > 13 11

Shift Operators

Two bit-wise operator are Shift left (<<) and shift right (>>) operators. Each operator operates on a single variable but requires two operands.

Shift Left operator (<<)

The shift left operator causes all the bits in the first operand to be shifted to the left by the number of position indicated by the second operand. The left most bits in the original bit pattern is lost. The right most bit position that become vacant are to be filled with zeros.

A = 1010 0110 -> 10 6

A<<2 = 1001 1000 -> 9 8

Shift Right Operator (>>)

The shift right operator causes all the bits in the first operand to be shifted to the right by the number of position indicated by the second operand. The right most bits in the original bit pattern is lost. The left most bit position that become vacant are to be filled with zeros.

A = 1010 0110 -> 10 6

A>>2 = 0010 1001 -> 2 9

Masking

Masking is a process in whicha given pattern is transformed into another bit pattern by means of logical bit – wise operation. The original bit pattern is one of the operands in the bit wise operation. The second operand *mask*, is a specially selected bit pattern that bring about the desired transformation.

BIT-WISE OPERATORS USED IN GA

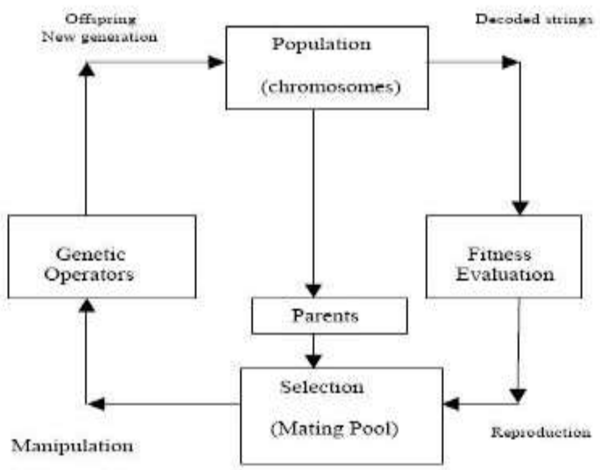
Logical bit-wise operators are used in different combination.Each operator operate on two individuals and generates one resultant so as to keep the number of individuals in the population constant. Two different operators are used in GA process.

CONVERGENCE OF GENETIC ALGORITHM

The convergence criteria can be explained from the schema point of view in the lines of Goldberg. A *schema* is a similarity template describing a subset of strings with similarities at certain position. Schema represent a subset of all possible strings that have the same string position.

Example:

Schema **000 represent the strings 00000, 01000, 10000 and 11000. Similarly a schema 1*00* to represent schema 10000, 10001, 11000 and 11001. Each string represented by a schema is called *instance of schema.* The symbol * signifies that a 0 or 1 could occurs the string position. Thus, the schema *****represent all possible string five bits. A schema average fitness value varies with the population's combination of from one generation to another. The GA cycle shown the fig



Building blocks are combined together due to combined action of genetic operator to form bigger and better building block and finally convergence to the optimal solution.

APPLICATIONS:

Genetic algorithms have been applied in science, engineering, business and socialsciences. Number of scientists has already solved many engineering problems using genetic algorithms. GA concepts can be applied to the engineering problem such as optimization of gas pipeline systems. Another important current area is structure optimization. The main objective in this problem is to minimize the weight of the structure subjected to maximum and minimum stress constrains on each member. GA is also used in medical imaging system. The GA is used to perform image registration as a part of larger digital subtraction angiographies. It can be found that Genetic Algorithm can be used over a wide range of applications. In this chapter a few topics of its application are being covered. This includes the application of Genetic Algorithm in to main engineering applications, data mining and in various other image processing applications. Hope the chapter would give the reader a brief idea of how the genetic algorithm can be applied to any practical problems.

Mechanical Sector

Optimizing Cyclic-Steam Oil Productionwith Genetic Algorithms

The Antelope reservoir in the Cymric field, in the San Joaquin Valley, is a siliceous shale reservoir containing 12 to 13◦API heavy oil. The reservoir consists primarily of diatomite, characterized by its high porosity, high oil saturation, and very low permeability. Approximately 430 wells are producing from this reservoir, with an

Average daily production of 23,000 bbl. The oil from the field is recovered using a Chevron-patented cyclic-steam process. A fixed amount of saturated steam is injected into the reservoir during a 3- to 4-day period. The high-pressure steam fractures the rock, and the heat from the steam reduces oil viscosity. The well is317 318 10 Applications of Genetic Algorithms shut in during the next couple of days, known as the soak period. Condensed steam is absorbed by the diatomite, and oil is displaced into the fractures and wellbore. After the soak period, the well is returned to production.

The flashing of hot water into steam at the prevailing pressure provides the energy to lift the fluids to the surface. The well flows for approximately 20 to 25 days. After the well dies, the same cycle is repeated. Cycle length is 26 to 30 days. Because there is no oil production during the steaming and soaking period, there is an incentive to minimize the steaming frequency and increase the length of the cycle. But because well production is highest immediately after returning to production and declines quickly thereafter, a case can be made for increasing the steaming frequency and reducing the length of the cycle.

This suggests that there is an optimum cycle length for every well that results in maximum productivity during the cycle. Because there are more than 400 wells in the field, and there are constraints of steam availability and distribution system, as well as facility constraints, the result is a formidable scheduling problem.

Genetic Algorithms

Genetic algorithms (GAs) are global optimization techniques developed by John Holland in 1975. They are one of several techniques in the family of evolutionary algorithms—algorithms that search for solutions to optimization problems by “evolving” better and better solutions. A genetic algorithm begins with a “population” of solutions and then chooses “parents” to reproduce. During reproduction, each parent is copied, and then parents may combine in an analog to natural crossbreeding, or the copies may be modified, in an analog to genetic mutation. The newnsolutions are evaluated and added to the population, and low-quality solutions are deleted from the population to make room for new solutions. As this process of parent selection, copying, crossbreeding, and mutation is repeated, the members of the population tend to get better. When the algorithm is halted, the best member of the current population is taken as the solution to the problem posed.

One critical feature of a GA is its procedure for selecting population members toreproduce. Selection is a random process, but a solution member’s quality biases its probability of being chosen. Because GAs promote the reproduction of high-quality solutions, they explore neighbouring solutions in high-quality parts of the solution search space. Because the process is randomized, a GA also explores parts of the search space that may be far from the best individuals currently in the population. In the last 20 years, GAs have been used to solve a wide range of optimization problems. There are many examples of optimization problems in the petroleum industry for which GAs are well suited. At ChevronTexaco, in addition to the cyclical steambscheduling problem, well placement, rig scheduling, portfolio optimization, and facilitiesdesign have been addressed with GAs. At NuTech Solutions, GAs have been used in planning rig workover projects so that overall workover time is reduced, planning production across multiple plants to reduce costs, planning distribution from multiple plants to a large number of customers to reduce costs, and controlling pipeline operations to reduce costs while satisfying pipeline constraints.