innovate achieve lead

BITS Pilani

# Machine Learning (IS ZC464) Session 15: Problem Solving and Review Session

# Problem 1: Classification using neural networks

- Consider a binary classification problem with two classes A and B
- Consider the following training data

<1,2,A> <7,11,B> <2,2,A>, <4,3,A>, <6,12,B>

- Design of a neural network appropriately and classify the test input feature vector <3,3> . Draw and annotate your neural network.

# Solution

Neural network design should focus on

- Establish if data is linearly separable
- output modeling,
- number of layers,
- number of processing neurons,
- Number of input and output neurons in each layer,
- activation function(s) and
- the weights

# Establish if data is linearly separable

- Plot the data on a 2 D Cartesian plane

# Output modeling

- Number of classes = 2
- How many bits can represent the class labels?

Answer: Only 1, i.e. if y=0, it is class A or if y = 1, it is class B

# Number of layers

- If data is linearly separable, then use 1 hidden layer

# Number of processing neurons

- How many linear decision boundaries are required?
- In this problem 2 decision boundaries are required and since the data is linearly separable, only lines are used as decision boundaries
- Only 2 processing neurons are required

# Number of input and output neurons in each layer

- Number of output neurons: one as discussed
- For number of input neurons, look at the feature vector size
- <1,2,A> <7,11,B> <2,2,A>, <4,3,A>, <6,12,B>
- It is 2.
- Hence number of input neurons = 2

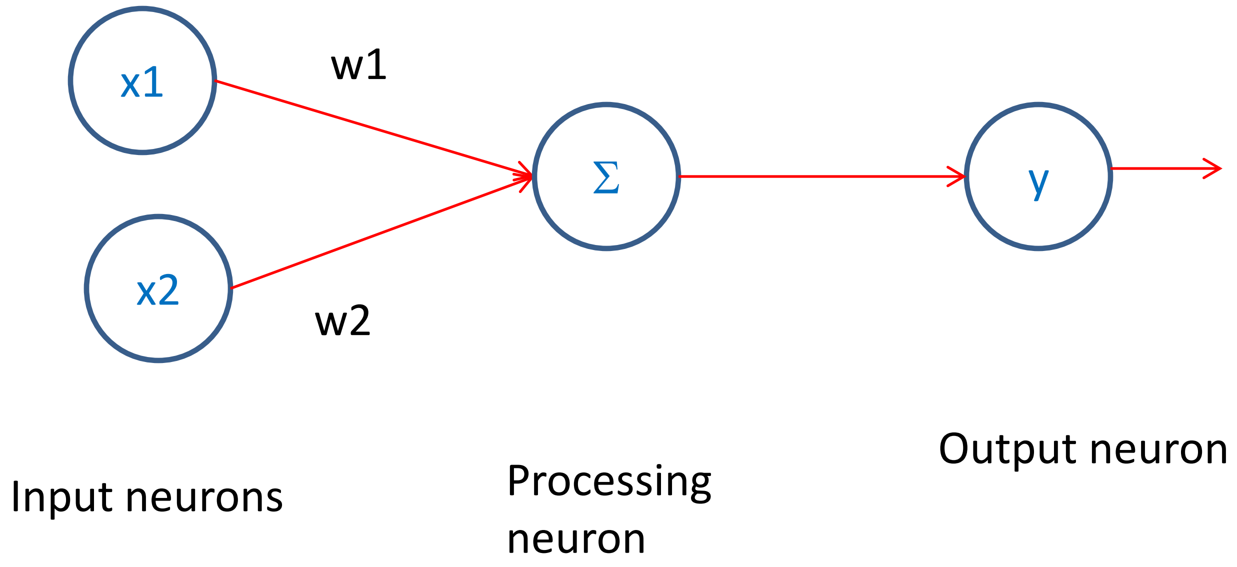
# activation function

- Step function if discrete valued output is expected
- g(weighted sum of inputs say n) = 1 if n>T

  0 otherwise
- What should be the threshold?

# Weights

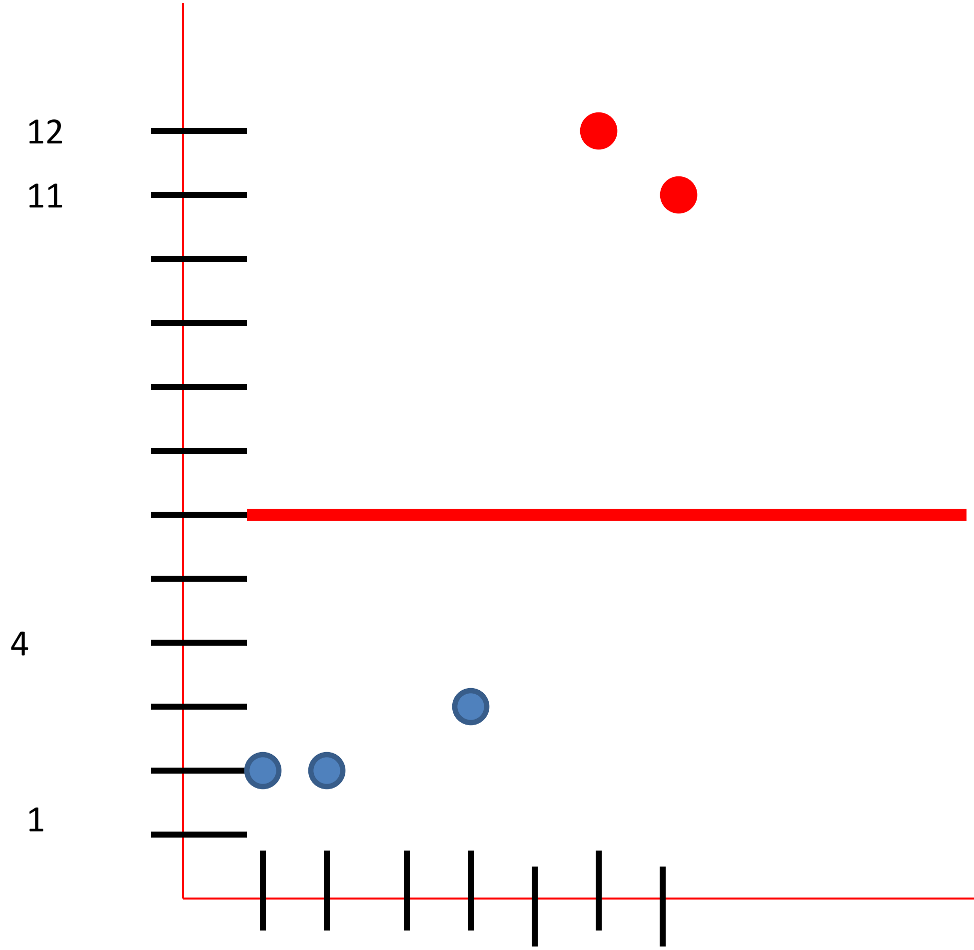
- How to compute weights?
- How many weights are needed?
- Draw a neural network for some idea.



# How to get the weights?

- Method 1: Approximation
- Method 2: parameters of the decision line
- Method 3: gradient descent
- Method 4: Genetic algorithm

# Using method 2 for weight computation




Training data
<1,2,A> <7,11,B>
<2,2,A>, <4,3,A>,
<6,12,B>

Draw a decision boundary

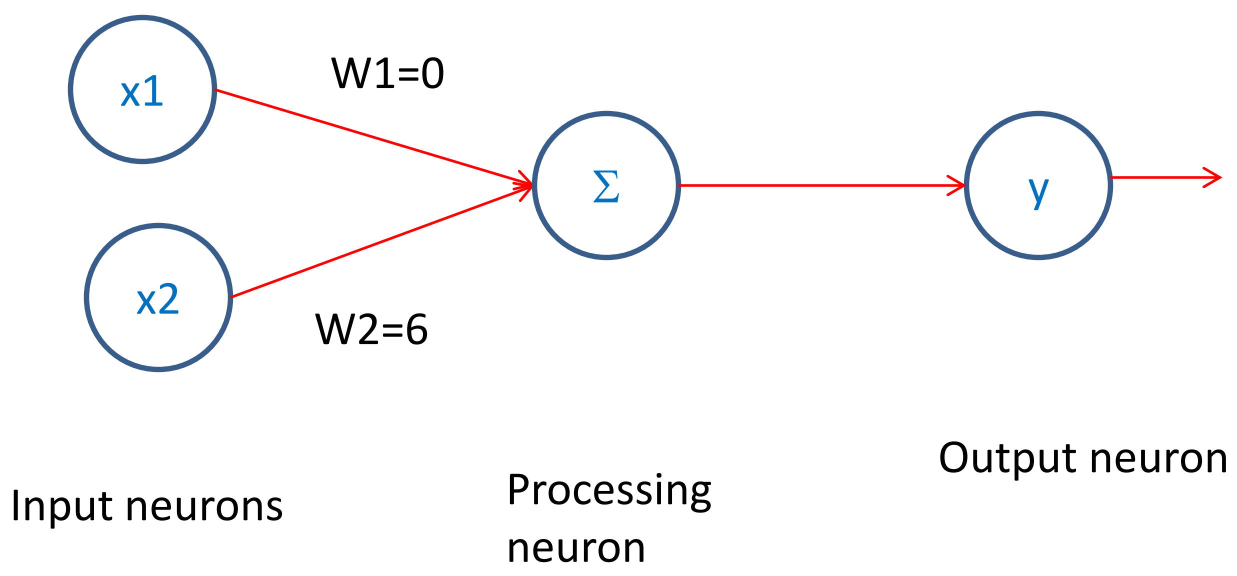
Equation y = 6
This means m =0
and c = 6
i.e. w1 = 0 w0 = 6

- <1,2,A> <7,11,B> <2,2,A>, <4,3,A>, <6,12,B>
- For feature vectors

<1,2> , the weighted sum n = 1*0+2*6 = 12 <T

<7,11> , n = 7*0+11*6 = 66 >T

Understand that threshold should be any number between 19 and 65.

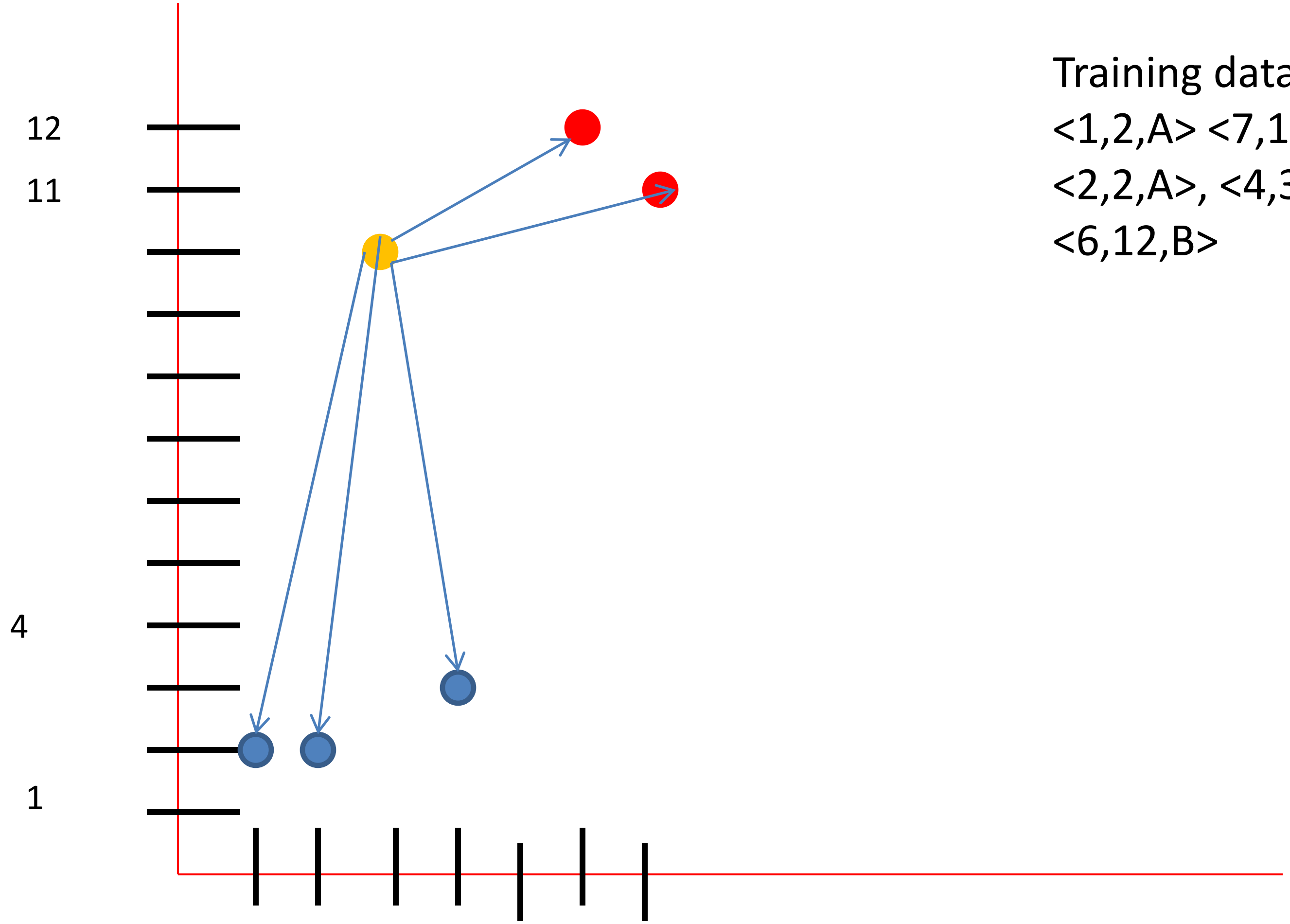


# Problem 2: Classification using k-nearest neighbors

- Consider a binary classification problem with two classes A and B
- Consider the following training data

<1,2,A> <7,11,B> <2,2,A>, <4,3,A>, <6,12,B>

# Take k = 3 and classify <3,10>




Training data
<1,2,A> <7,11,B>
<2,2,A>, <4,3,A>,
<6,12,B>

# Problem 3: Genetic algorithm

- Given a sequence of numbers

23, 45, 11, 10, 2, 101, 1

- Design a genetic algorithm based solution to find the maximum of given numbers

- Complexity of the problem is same as that of searching a list of n numbers to find maximum value.

# Chromosome formulation

- Let us define a feasible solution as a bit string

  $a_6a_5a_4a_3a_2a_1a_0$

  where $a_i$ is a binary digit having a value equal to 1 or 0 indicating whether an element at $i^{th}$ position **from right** is in the given sequence is included in the bit string or not.
- Example bit string 0011001 is a string that includes from sequence 23, 45, 11, 10, 2, 101, 1 the elements 1, 10 and 11

# Fitness function formulation

- To find out maximum we compute the maximum of the numbers 1, 10 and 11 (for example) which is 11.
- i.e. fitness function is given by

  $f(a_6a_5a_4a_3a_2a_1a_0)$ = Maximum of numbers whose representational bits are 1s

# Initial population

- For the sequence 23, 45, 11, 10, 2, 101, 1 we generate 5 initial chromosome population as follows

| Chromosome bit string | Elements included | Fitness value |
|---|---|---|
| 0010100 | 2, 11 | 11 |
| 1010101 | 1,2,11,23 | 23 |
| 0010101 | 1,2,11 | 11 |
| 1010010 | 101,11,23 | 101 |
| 0000110 | 101,2 | 101 |

# Pool of fit parents

- A pool of parent chromosomes with threshold value say T= 5 is considered in this example
- Then all initial parents are fit to be included in the parent selection

# Parent selection: roulette wheel

| S. No. | Chromosome bit string | Elements included | Fitness value | probability | Cumulative frequency |
|---|---|---|---|---|---|
| 1 | 0010100 | 2, 11 | 11 | 11/247=0.045 | 0.045 |
| 2 | 0010101 | 1,2,11 | 11 | 11/247=0.045 | 0.090 |
| 3 | 1010101 | 1,2,11,23 | 23 | 23/247=0.093 | 0.183 |
| 4 | 1010010 | 101,11,23 | 101 | 101/247=0.4085 | 0.5915 |
| 5 | 0000110 | 101,2 | 101 | 101/247=0.4085 | 1.0000 |

Use random number generator and generate two numbers between 0 and 1

If rand() = 0.3, then select 4th bit string
If rand() = 0.017, then select 1st bit string

# Crossover and mutation probability

- Crossover probability $P_c$: to select parents to produce new offsprings by cross over operation
- Mutation probability $P_m$ :to select randomly the bits from each bit string for mutation operation

# Crossover and mutation probability

- If given $p_c = 0.8$
- This means that out of 5 potential parent chromosomes we will select 4 using roulette wheel method.

- And $p_m = 0.3$
- This means that out of 7 bits, about 2 bits (as against a number 2.1) will be mutated

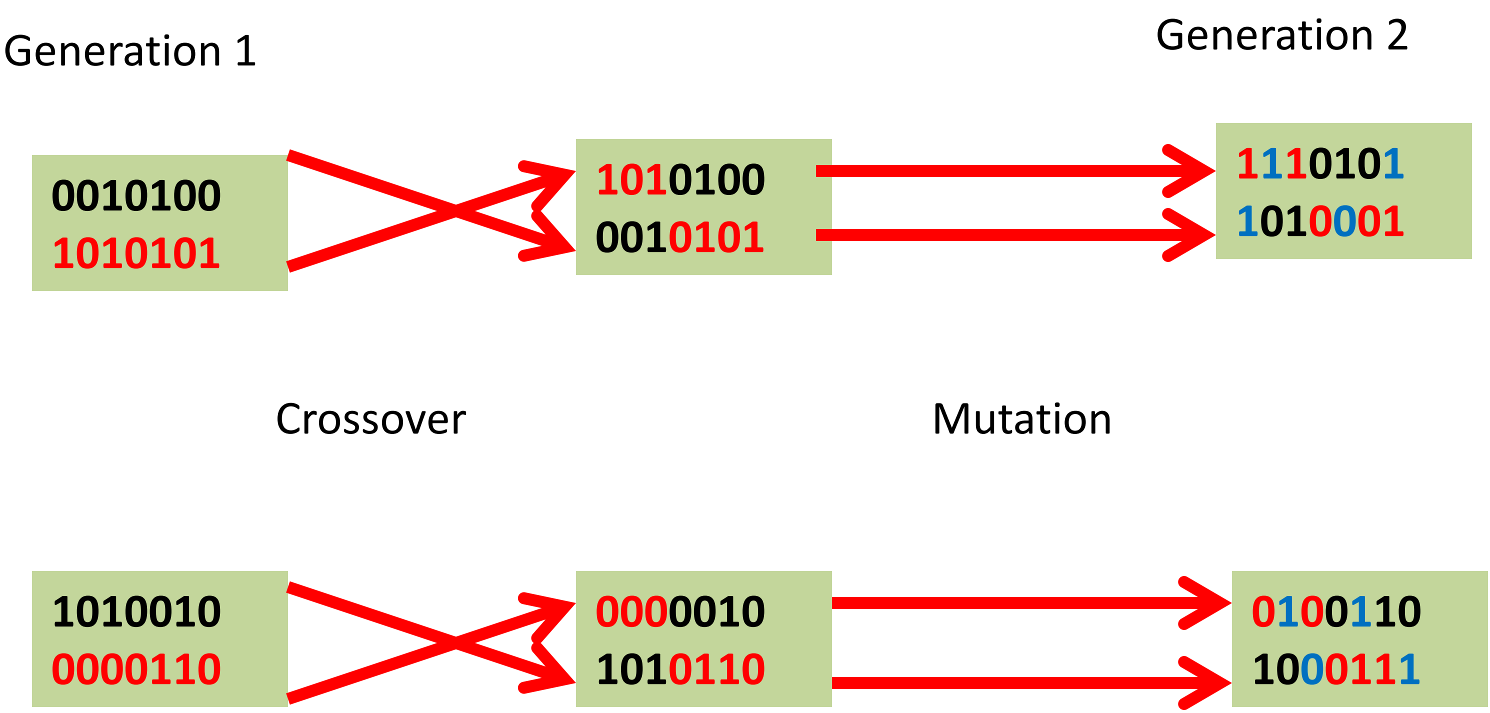
# Parent selection: 4 parents

| S. No. | Chromosome bit string | Elements included | Fitness value | probability | Cumulative frequency |
|---|---|---|---|---|---|
| 1 | 0010100 | 2, 11 | 11 | 11/247=0.045 | 0.045 |
| 2 | 0010101 | 1,2,11 | 11 | 11/247=0.045 | 0.090 |
| 3 | 1010101 | 1,2,11,23 | 23 | 23/247=0.093 | 0.183 |
| 4 | 1010010 | 101,11,23 | 101 | 101/247=0.4085 | 0.5915 |
| 5 | 0000110 | 101,2 | 101 | 101/247=0.4085 | 1.0000 |

If rand() = 0.3, then select 4th bit string
If rand() = 0.017, then select 1st bit string

If rand() = 0.629, then select 5th bit string
If rand() = 0.17, then select 3rd bit string

# Crossover and mutation operations using one point crossover between 3rd and 4th bit



# Re-compute fitness: given sequence 23, 45, 11, 10, 2, 101, 1

| Generation 1 | Elements included | Fitness value |
|---|---|---|
| 0010100 | 2, 11 | 11 |
| 1010101 | 1,2,11,23 | 23 |
| 1010010 | 101,11,23 | 101 |
| 0000110 | 101,2 | 101 |

Average fitness = 236/4 = 59

| Generation 2 | Elements included | Fitness value |
|---|---|---|
| **1110101** | 1, 2, 11, 45, 23 | 45 |
| **1010001** | 1, 11, 23 | 23 |
| **0100110** | 101, 2, 45 | 101 |
| **1000111** | 1,101,2,23 | 101 |

Average fitness = 270/4 = 67.5

# Session 1: Introduction

- Learning – experience from past
- Learning -- humans vs. machines
- Why machine learning?
- Machine learning and Artificial intelligence
- Intelligent systems
- Examples – vehicle recognition, fruit recognition and face recognition: attributes and their significance
- Traditional vs. Machine learning
- etc.

# Session 2: Regression

- Training and testing in learning systems
- Prediction – training and testing
- (X,Y) – input output pair
- Understanding error
- Function approximation : given training (x,y) pairs
- Best hypothesis-minimum error tradeoff
- Learning parameters defining a hypothesis
- Line fitting – slope and intercept
- Squared mean error
- etc.

# Session 3: Classification

- Decision boundaries and regions
- Binary classification
- Linear separability
- Discriminant functions
- Identifying the decision boundary on 2D plane
- Classifying a test vector using $h(x) = x_2 - mx_1 - c$
- etc.

# Session 4: Classification Algorithms

- K-Nearest Neighbors
- Lazy learning
- Distance based weighted k-NN algorithm
- Bayes' Theorem
- Naïve Bayes' Classifier
- Problem solving- classifying a test pattern
- etc.

# Session 5: Decision tree and data visualization

- Decision tree classifier
- Decision tree and Its limitations
- Attributes and their significance in decision tree based classification
- Leaf and non leaf nodes of a decision tree
- Information content – entropy of the given set of observations
- Splitting the observations based on decision key and attribute selected for splitting
- Gain and remainder computations
- Data visualization
- Scatter plots
- Parallel coordinate graphs
- Pearson Correlation coefficient
- etc.

# Session 6: Feature Engineering

- What is feature engineering
- Machine learning model with feature engineering
- Dimensionality reduction
- Feature Extraction techniques
- Bag of words
- TF-IDF, n-grams, stemming
- Image features, glimpse of transform based features
- Feature selection algorithms etc.

# Session 7: Evaluation of Classification models

Overfitting

Bias and variance

Partitioning the data

Hold out, Cross validation and bootstrapping methods

Class-wise accuracy

true/false positives/negatives

Precision and recall

sensitivity analysis

ROC curves

Confusion matrix etc.

# Session 8: Review

# Session 9: Neural Networks

- Human nervous system : some inspiration
- Artificial neuron
- Simulation  of a single neuron to perform logical AND-weight, threshold, weighted input, activation function, training and testing, unseen patterns for testing
- XOR simulation, why non linearly separable data, why two neurons, threshold, weights, activation functions for the required neurons
- etc.
- Neural network simulation for XOR and related computations
- Fruit recognition example and it neural network simulation-output modeling, weights etc
- Feed forwards neural network architecture
- Significance of layers
- Learning factors
- etc.

# Session 10 : Neural Networks

- Error surface for Neural Network based classification
- Computing Gradient
- Delta rule of weight update in gradient descent method
- Learning rate
- Weights as knowledge – classification of unseen pattern using acquired weights
- Chain rule of partial derivatives – used for computing the delta of weight + understanding of the derivation of the term for weight update
- etc.

# Session 11: MLP and RBFNN

- Multilayer feed forward neural network
- Face recognition problem
- Radial basis function neural network

# Session 12: Genetic Algorithms

- Evolutionary algorithms and their applications
- Population based algorithm
- Chromosome as bit string
- Representing rules as bit strings
- Search as combinatorial problem
- Fitness value
- Crossover and mutation
- etc.

# Session 13: Genetic Algorithms

- Roulette wheel method
- Fittest parent-pool of parents
- Exploration and exploitation
- Application of genetic algorithm in unsupervised clustering, clustering as an optimization problem, fitness function for clustering
- Limitation of GA
- etc.

# Session 14: Support Vector Machines

- Support vector machines
- Support vectors – understanding
- Margin and decision boundary
- Computation of the margin
- What to optimize in SVM?
- Kernel functions
- etc.

# Session 15: Problem solving and review

# Syllabus for Comprehensive examination

- All topics and the corresponding depth of concepts covered during lecture sessions 1 to 15.