SYLLABUS

Machine Learning [CS601PC/IT523PE]

UNIT - I

Introduction - Well-posed learning problems, designing a learning system, Perspectives and issues in machine learning.

Concept learning and the general to specific ordering - introduction, a concept learning task, concept learning as search, find-S : finding a maximally specific hypothesis, version spaces and the candidate elimination algorithm, remarks on version spaces and candidate elimination, inductive bias.

Decision Tree Learning - introduction, decision tree representation, appropriate problems for decision tree learning, the basic decision tree learning algorithm, hypothesis space search in decision tree learning, inductive bias in decision tree learning, issues in decision tree learning. **(Chapter - 1)**

UNIT - II

Artificial Neural Networks - 1 - Introduction, neural network representation, appropriate problems for neural network learning, perceptions, multilayer networks and the back-propagation algorithm.

Artificial Neural Networks - 2 - Remarks on the Back-Propagation algorithm, An illustrative example : face recognition, advanced topics in artificial neural networks.

Evaluation Hypotheses - Motivation, estimation hypothesis accuracy, basics of sampling theory, a general approach for deriving confidence intervals, difference in error of two hypotheses, comparing learning algorithms. **(Chapter - 2)**

UNIT - III

Bayesian learning - Introduction, Bayes theorem, Bayes theorem and concept learning, Maximum Likelihood and least squared error hypotheses, maximum likelihood hypotheses for predicting probabilities, minimum description length principle, Bayes optimal classifier, Gibs algorithm, Naïve Bayes classifier, an example : learning to classify text, Bayesian belief networks, the EM algorithm.

Computational learning theory - Introduction, probably learning an approximately correct hypothesis, sample complexity for finite hypothesis space, sample complexity for infinite hypothesis spaces, the mistake bound model of learning.

Instance-Based Learning - Introduction, *k*-nearest neighbour algorithm, locally weighted regression, radial basis functions, case-based reasoning, remarks on lazy and eager learning. **(Chapter - 3)**

UNIT- IV

Genetic Algorithms - Motivation, Genetic algorithms, an illustrative example, hypothesis space search, genetic programming, models of evolution and learning, parallelizing genetic algorithms.

Learning Sets of Rules - Introduction, sequential covering algorithms, learning rule sets : summary, learning First-Order rules, learning sets of First-Order rules : FOIL, induction as inverted deduction, inverting resolution.

Reinforcement Learning - Introduction, the learning task, Q-learning, non-deterministic, rewards and actions, temporal difference learning, generalizing from examples, relationship to dynamic programming. **(Chapter - 4)**

UNIT - V

Analytical Learning - 1 - Introduction, learning with perfect domain theories : PROLOG-EBG, remarks on explanation-based learning, explanation-based learning of search control knowledge.

Analytical Learning - 2 - Using prior knowledge to alter the search objective, using prior knowledge to augment search operators.

Combining Inductive and Analytical Learning - Motivation, inductive-analytical approaches to learning, using prior knowledge to initialize the hypothesis. **(Chapter - 5)**

TABLE OF CONTENTS

Unit - I

Unit - II

Unit - III

Unit - IV

Unit - V

UNIT - I

1 Introduction

1.1 : Introduction : Well Posed Learning Problems

Q.1 Define learning. [JNTU : Dec.-17, Marks 2]

Ans. : Learning is a phenomenon and process which has manifestations of various aspects. Learning process includes gaining of new symbolic knowledge and development of cognitive skills through instruction and practice. It is also discovery of new facts and theories through observation and experiment.

Q.2 Define machine learning.

Ans. : A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Q.3 What is need of machine learning in this era ? [JNTU : Dec.-16, Marks 3]

Ans. : • Main goal of machine learning is to devise learning algorithms that do the learning automatically without human intervention or assistance.

- The machine learning paradigm can be viewed as "programming by example." Another goal is to develop computational models of human learning process and perform computer simulations.
- The goal of machine learning is to build computer systems that can adapt and learn from their experience.
- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples.
- Machine Learning provides business insight and intelligence. Decision makers are provided with greater insights into their organizations. This adaptive technology is being used by global enterprises to gain a competitive edge.
- Machine learning algorithms discover the relationships between the variables of a system (input, output and hidden) from direct samples of the system.

Q.4 What are T, P, E ? How do we formulate a machine learning problem ?

Ans. : • In general, to have a well-defined learning problem, we must identity these three features : the class of tasks, the measure of performance to be improved, and the source of experience.

- A Robot Driving Learning Problem
  1. Task T : Driving on public, 4-lane highway using vision sensors.
  2. Performance measure P : Average distance traveled before an error (as judged by human overseer).
  3. Training experience E : A sequence of images and steering commands recorded while observing a human driver.
- A Handwriting Recognition Learning Problem
  1. Task T : Recognizing and classifying handwritten words within images
  2. Performance measure P : Percent of words correctly classified.
  3. Training experience E : A database of handwritten words with given classifications.
- Text Categorization Problem
  1. Task T : Assign a document to its content category.
  2. Performance measure P : Precision and Recall.
  3. Training experience E : Example pre-classified documents.

Q.5 What are the reasons for using machine learning ?

Ans. : Following are some of the reasons :

1. Some tasks cannot be defined well, except by examples. For example : recognizing people.
2. Relationships and correlations can be hidden within large amounts of data. To solve these problems, machine learning and data mining may be able to find these relationships.
3. Human designers often produce machines that do not work as well as desired in the environments in which they are used.
4. The amount of knowledge available about certain tasks might be too large for explicit encoding by humans.
5. Environments change time to time.
6. New knowledge about tasks is constantly being discovered by humans.

Q.6 List the phases of machine learning ?

Ans. : Typically follows three phases :

1. **Training :** A training set of examples of correct behaviour is analysed and some representation of the newly learnt knowledge is stored. This is some form of rules.
2. **Validation :** The rules are checked and, if necessary, additional training is given. Sometimes additional test data are used, but instead, a human expert may validate the rules, or some other automatic knowledge - based component may be used. The role of the tester is often called the opponent.
3. **Application :** The rules are used in responding to some new situation.

Q.7 What is meant by machine learning ? What is its need to today's society ? Explain successful applications of machine learning ?

[JNTU : Dec.-17, Marks 10]

Ans. : • Examples of successful applications of machine learning :

1. Learning to recognize spoken words.
2. Learning to drive an autonomous vehicle.
3. Learning to classify new astronomical structures.
4. Learning to play world-class backgammon.
5. Spoken language understanding: within the context of a limited domain, determine the meaning of something uttered by a speaker to the extent that it can be classified into one of a fixed set of categories

Face Recognition

- Face recognition task is effortlessly and every day we recognize our friends, relative and family members. We also recognition by looking at the photographs. In photographs, they are in different pose, hair styles, background light, makeup and without makeup.
- We do it subconsciously and cannot explain how we do it. Because we can't explain how we do it, we can't write an algorithm.
- Face has some structure. It is not a random collection of pixel. It is symmetric structure. It contains predefined components like nose, mouth, eye, ears. Every person face is a pattern composed of a particular combination of the features. By analyzing sample face images of a person, a learning program captures the pattern specific to that person and uses it to recognize if a new real face or new image belongs to this specific person or not.
- Machine learning algorithm creates an optimized model of the concept being learned based on data or past experience.

Q.8 Explain the difference between machine learning and data mining.

Ans. :

Machine Learning	Data Mining
In machine learning the main goal is to learn a model, which can be used to predict future events.	In data mining, the main goal is to discover new interesting information which describes the current data set.
It considered data as secondary.	It considered data as primary.
Machine learning uses relatively complex and global models.	Data mining uses simple models or local patterns.

Only hundreds or thousands of examples in a training data set.	Huge data sets, even millions of rows.
To learn one or few carefully defined models, this can be used to predict future events.	To find all interesting patterns which describe the data set.

Q.9 What are the ingredients of machine learning ?

Ans. : The ingredients of machine learning are as follows :

1. **Tasks :** The problems that can be solved with machine learning. A task is an abstract representation of a problem. The standard methodology in machine learning is to learn one task at a time. Large problems are broken into small, reasonably independent sub-problems that are learned separately and then recombined.
   - Predictive tasks perform inference on the current data in order to make predictions. Descriptive tasks characterize the general properties of the data in the database
2. **Models :** The output of machine learning. Different models are geometric models, probabilistic models, logical models, grouping and grading.
   - The model-based approach seeks to create a modified solution tailored to each new application. Instead of having to transform your problem to fit some standard algorithm, in model-based machine learning you design the algorithm precisely to fit your problem.
   - Model is just made up of set of assumptions, expressed in a precise mathematical form. These assumptions include the number and types of variables in the problem domain, which variables affect each other, and what the effect of changing one variable is on another variable.
   - Machine learning models are classified as : Geometric model, Probabilistic model and Logical model.
3. **Features :** The workhorses of machine learning. A good feature representation is central to achieving high performance in any machine learning task.
   - Feature extraction starts from an initial set of measured data and builds derived values intended to be informative, non redundant, facilitating the subsequent learning and generalization steps.
   - Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion.

1.2 : Designing Learning System

Q.10 What is an influence of information theory on machine learning ? [JNTU : Dec.-17, Marks 3]

Ans. : Information theory is measures of entropy and information content. Minimum description length approaches to learning. Optimal codes and their relationship to optimal training sequences for encoding a hypothesis.

Q.11 What is meant by target function of a learning program ? [JNTU : Dec.-16, Marks 2]

Ans. : Target function is a method for solving a problem that an AI algorithm parses its training data to find. Once an algorithm finds its target function, that function can be used to predict results. The function can then be used to find output data related to inputs for real problems where, unlike training sets, outputs are not included.

Q.12 Explain basic design issue and approaches of machine learning.

Ans. : Designing a Learning System

- Goal : Design a system to learn how to play checkers and enter it into the world checkers tournament.
  1) Choose the training experience
  2) Choose the target function
  3) Choose a representation for the target function
  4) Choose a function approximation algorithm

Training Experience

- How training experience influences performance goal ?
  1. Type of feedback : Direct vs Indirect.
  2. Learning strategy : Have a teacher or not ? Exploration vs Exploitation ?

3. Diversity of training : Is the training data representative of the task ? How many peers should we play with? How many tactics should we try when playing with self ?

- Let us decide that our program will learn by playing with itself and formulate the learning problem.
- Choosing the training experience :
  1. Direct or indirect feedback
  2. Degree of learner's control
  3. Representative distribution of examples
- **Learning Goal is to :** Define precisely a class of problems that forms interesting forms of learning, explore algorithms to solve such problems, understand fundamental structure of learning problems and processes.
- Design choice 1 : The problem for selecting type of training experience from which our system will learn. Direct training examples. Just a bunch of board states together with a correct move.
- Design Choice 2 : Indirect training. A bunch of recorded games, where the correctness of the moves is inferred by the result of the game.
- Learning is most reliable when the training examples follow a distribution similar to that of future test examples.

Q.13 How to choose and represent the target function ?

Ans. : • It determines exactly what type of knowledge will be learned and how this will be used by the performance program.

- Choosing a representation for the target function :
  1. Expressive representation for a close function approximation
  2. Simple representation for simple training data and learning algorithms

$$V(b) = w_0 + w_1 X_1 + \ldots + w_6 X_6$$

$X_{1,2}$: Number of black/red pieces on the board

$X_{3,4}$: Number of black/red kings on the board

$X_{5,6}$: Number of black/red pieces threatened can be captured on red/black next turn)

- Consider the chess board program.
- ChooseMove : B -> M where B is any legal board state and M is a legal move (hopefully the "best" legal move)
- Alternatively, function V : B -> $\Re$ which maps from B to some real value where higher scores are assigned to better board states.
- Now use the legal moves to generate every subsequent board state and use V to choose the best one and therefore the best legal move.
  1. V(b) = 100, if b is a final board state that is won
  2. V(b) = –100, if b is a final board state that is lost
  3. V(b) = 0, if b is a final board state that is a draw
  4. V(b) = V(b'), if b is not a final state where b' is the best final board state starting from b assuming both players play optimally
- While this recursive definition specifies a value of V(b) for every board state b, this definition is not usable by our checkers player because it is not efficiently computable.
- For representation
  1. Use a large table with an entry specifying a value for each distinct board state.
  2. Collection of rules that match against features of the board state.
  3. Quadratic polynomial function of predefined board features.

Q.14 How to adjust the weights ?

Ans. : • Choose the weights w_i to best fit the set of training examples.

- Minimize the squared error E between the train values and the values predicted by the hypothesis

$$E = \sum_{(b, V_{train}(b)) \in \text{training examples}} (V_{train}(b) - \hat{V}(b))^2$$

- Require an algorithm that will incrementally refine weights as new training examples become available and it will be robust to errors in these estimated training values.
- Least Mean Squares (LMS) is one such algorithm.

1.3 : Perspectives and Issues in Machine Learning

Q.15 Define useful perspective on machine learning.

Ans. : One useful perspective on machine learning is that it involves searching a very large space of possible hypotheses to determine one that best fits the observed data and any prior knowledge held by the learner

Q.16 Describe the issues in machine learning ?

Ans. : Issues of machine learning are as follows :

- What learning algorithms to be used ?
- How much training data is sufficient ?
- When and how prior knowledge can guide the learning process ?
- What is the best strategy for choosing a next training experience ?
- What is the best way to reduce the learning task to one or more function approximation problems ?
- How can the learner automatically alter its representation to improve its learning ability ?

1.4 : Concept Learning and the General to Specific Ordering

Q.17 What is hypothesis ?

Ans. : • A hypothesis is a vector of constraints for each attribute

1. Indicate by a "?" that any value is acceptable for this attribute
2. Specify a single required value for the attribute
3. Indication by a "Ø" that no value is acceptable

- If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example (h(x) = 1).

Q.18 What is the inductive learning hypothesis ?

Ans. : Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

Q.19 Explain with example concept learning task.

Ans. : • Inducing general functions from specific training examples is a main issue of machine learning.

- Concept Learning: Acquiring the definition of a general category from given sample positive and negative training examples of the category.
- Concept Learning can be seen as a problem of searching through a predefined space of potential hypotheses for the hypothesis that best fits the training examples.
- The hypothesis space has a general-to-specific ordering of hypotheses, and the search can be efficiently organized by taking advantage of a naturally occurring structure over the hypothesis space.
- Formal Definition for Concept Learning: Inferring a boolean-valued function from training examples of its input and output.
- An example for concept-learning is the learning of bird-concept from the given examples of birds (positive examples) and non-birds (negative examples).

- We are trying to learn the definition of a concept from given examples.
- Concept learning involves determining a mapping from a set of input variables to a Boolean value. Such methods are known as inductive learning methods.
- If a function can be found which maps training data to correct classifications, then it will also work well for unseen data. This process is known as generalization.
- Example : Learn the "days on which my friend enjoys his favorite water sport"

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
1	Sunny	Warm	Normal	Strong	Warm	Same	YES
2	Sunny	Warm	High	Strong	Warm	Same	YES
3	Rainy	Cold	High	Strong	Warm	Change	NO
4	Sunny	Warm	High	Strong	Warm	Change	YES
	ATTRIBUTES						↑ CONCEPT

- A set of example days, and each is described by six attributes. The task is to learn to predict the value of EnjoySport for arbitrary day, based on the values of its attribute values.
- **The inductive learning hypothesis :** Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.
- Although the learning task is to determine a hypothesis (h) identical to the target concept cover the entire set of instances (X), the only information available about c is its value over the training examples.
- Inductive learning algorithms can at best guarantee that the output hypothesis fits the target concept over the training data.
- Lacking any further information, our assumption is that the best hypothesis regarding unseen instances is the hypothesis that best fits the observed training data. This is the fundamental assumption of inductive learning.
- Hypothesis representation (constraints on instance attributes) :
  <Sky, AirTemp, Humidity, Wind, Water, Forecast>

1. Any value is acceptable is represented by ?
2. No value is acceptable is represented by Φ

Q.20 Illustrate general to specific ordering of hypotheses in concept learning ? ☞ **[JNTU : Dec.-17, Marks 5]**

Ans. :
- Many algorithms for concept learning organize the search through the hypothesis space by relying on a general-to-specific ordering of hypotheses.
- By taking advantage of this naturally occurring structure over the hypothesis space, we can design learning algorithms that exhaustively search even infinite hypothesis spaces without explicitly enumerating every hypothesis.

- Consider two hypotheses:

  h1 = (Sunny, ?, ?, Strong, ?, ?)

  h2 = (Sunny, ?, ?, ?, ?, ?)

- Now consider the sets of instances that are classified positive by hl and by h2. Because h2 imposes fewer constraints on the instance, it classifies more instances as positive.
- In fact, any instance classified positive by hl will also be classified positive by h2. Therefore, we say that h2 is more general than hl.
- One learning method is to determine the most specific hypothesis that matches all the training data.
- More-General-Than-Or-Equal Relation : Let h1 and h2 be two boolean-valued functions defined over X. Then h1 is more-general-than-or-equal-to h2 (written h1 ≥ h2). If and only if any instance that satisfies h2 also satisfies h1.
- h1 is more-general-than h2 (h1 > h2) if and only if h1 ≥ h2 is true and h2 ≥ h1 is false. We also say h2 is more-specific-than h1.

$h_j \geq h_k$ iff $\forall x \in X : h_k(X) = 1 \Rightarrow h_j(x) = 1$

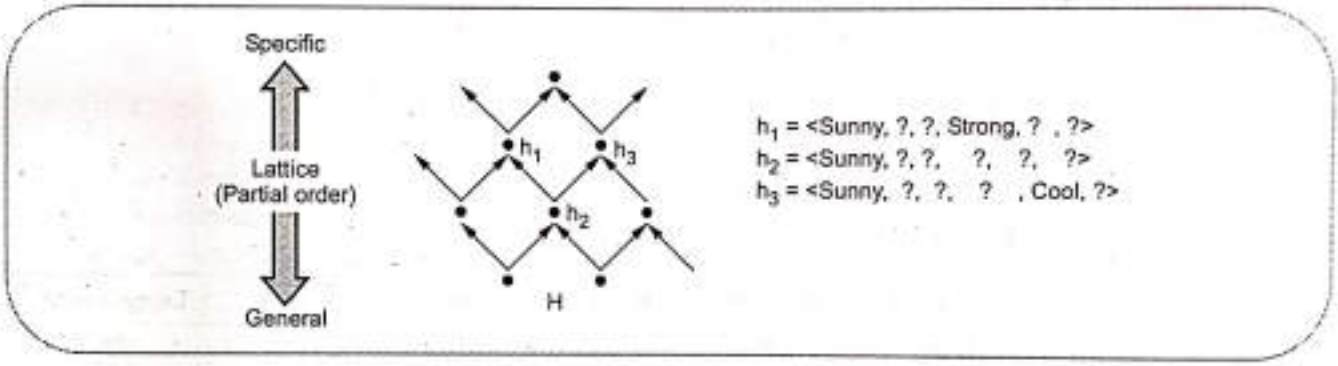


Fig. Q.20.1

1.5 : FIND-S Algorithm

Q.21 Explain the key properties of FIND-S algorithm for concept learning with necessary example.

☞ [JNTU : Dec.-17, Marks 5]

Ans. : • The key property is that for hypothesis spaces described by conjunctions of attribute constraints, FIND-S is guaranteed to output the most specific hypothesis within H that is consistent with the positive training examples.

- Its final hypothesis will also be consistent with the negative examples provided the correct target concept is contained in H, and provided the training examples are correct.
- FIND-S Algorithm starts from the most specific hypothesis and generalize it by considering only positive examples.
- This algorithm ignores negative examples. As long as the hypothesis space contains a hypothesis that describes the true target concept, and the training data contains no errors, ignoring negative examples does not cause to any problem.
- FIND-S algorithm finds the most specific hypothesis within H that is consistent with the positive training examples.

- The final hypothesis will also be consistent with negative examples if the correct target concept is in H, and the training examples are correct.

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
1	Sunny	Warm	Normal	Strong	Warm	Same	YES
2	Sunny	Warm	High	Strong	Warm	Same	YES
3	Rainy	Cold	High	Strong	Warm	Change	NO
4	Sunny	Warm	High	Strong	Cool	Change	YES

h = < ϕ, ϕ, ϕ, ϕ, ϕ, ϕ>

h = <Sunny, Warm, Normal, Strong, Warm, Same>

h = <Sunny, Warm, ?, Strong, Warm, Same>

h = <Sunny, Warm, ?, Strong, ?, ?>

Algorithm :

- Initialize h to the most specific hypothesis in H :
- For each attribute training instance x :

  For each attribute constraint a in h :

  If the constraint is not satisfied by x.

  Then replace a, by the next more general constraint satisfied by x.
- Output hypothesis h

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
1	Sunny	Warm	Normal	Strong	Warm	Same	YES
2	Sunny	Warm	High	Strong	Warm	Same	YES
3	Rainy	Cold	High	Strong	Warm	Change	NO
4	Sunny	Warm	High	Strong	Cool	Change	YES

h = <Sunny, Warm, ?, Strong, ?, ?>

Prediction :

5	Rainy	Cold	High	Strong	Warm	Change	NO
6	Sunny	Warm	Normal	Strong	Warm	Same	YES
7	Sunny	Warm	Low	Strong	Cool	Same	YES

Q.22 Describe hypothesis space search by FIND-S algorithm. [JNTU : Dec.-16, Marks 10]

Ans. : • The FIND-S algorithm illustrates one way in which the more-general than partial ordering can be used to organize the search for an acceptable hypothesis.

- The search moves from hypothesis to hypothesis, searching from the most specific to progressively more general hypotheses along one chain of the partial ordering.
- The hypothesis space search performed by FIND-S.

- The search begins (h_0) with the most specific hypothesis in H, then considers increasingly general hypotheses (h_1 through h_4) as mandated by the training examples.
- In the instance space diagram, positive training examples are denoted by "+," negative by "-," and instances that have not been presented as training examples are denoted by a solid circle.
- At each stage the hypothesis is the most specific hypothesis consistent with the training examples observed up to this point.

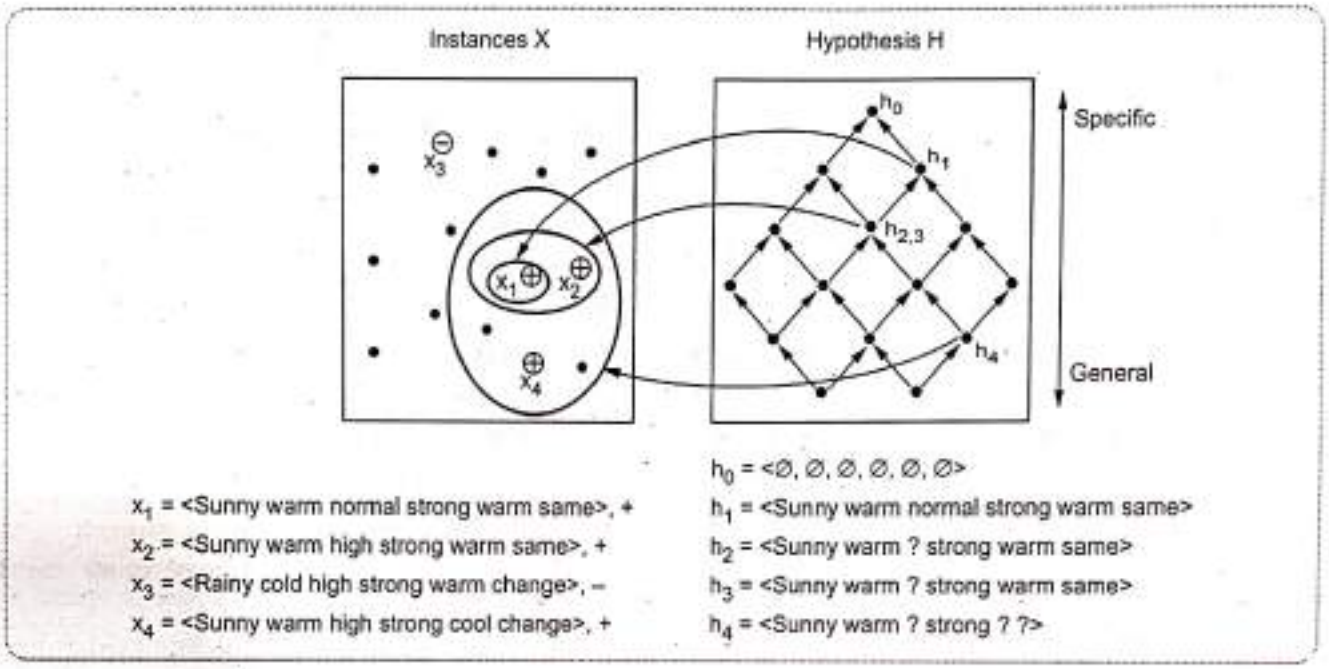


Fig. Q.22.1

1.6 : Version Space and Candidate Elimination Algorithm

Q.23 What is version space ?

Ans. :
- **Version space :** A set of all hypotheses that are consistent with the training examples.
- The **version space**, denoted VS_H,D, with respect to hypothesis space H and training examples D, is the subset of from H consistent with the training examples in D.
- A version space is a hierarchical representation of knowledge that enables you to keep track of all the useful information supplied by a sequence of learning examples without remembering any of the examples.
- The version space method is a concept learning process accomplished by managing multiple models within a version space.

Q.24 Explain characteristics of version space.

Ans. : **Characteristics :**

1. Tentative heuristics are represented using version spaces.
2. A version space represents all the alternative plausible descriptions of a heuristic.
3. A plausible description is one that is applicable to all known positive examples and no known negative example.

4. A version space description consists of two complementary trees :
   i. One that contains nodes connected to overly general models, and
   ii. One that contains nodes connected to overly specific models.
5. Node values/attributes are discrete.

Q.25 Describe compact representation for version spaces.

Ans. :

- Instead of enumerating all the hypotheses consistent with a training set, we can represent its most specific and most general boundaries. The hypotheses included in-between these two boundaries can be generated as needed.
- Definition: The general boundary (G) with respect to hypothesis space H and training data D, is the set of maximally general members of H consistent with D.
- Definition : The specific boundary (S) with respect to hypothesis space H and training data D, is the set of minimally general (i.e., maximally specific) members of H consistent with D.
- Fig. Q.25.1 shows general and specific hypothesis.

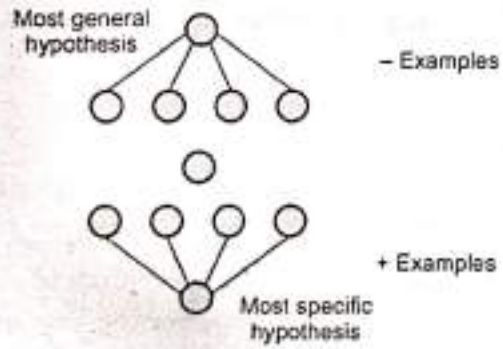


Fig. Q.25.1 General and specific hypothesis

- Each specialization must be a generalization of some specific concept description. No specialization can be a specialization of another general concept description. Fig. Q.25.2 shows boundary set with hypothesis

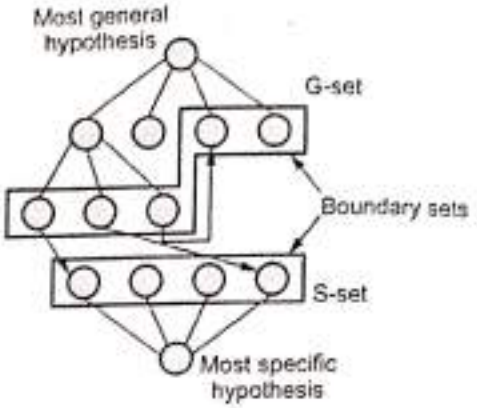


Fig Q.25.2 Boundary set with hypothesis

Q.26 What are advantages and disadvantages of version space method ?

Ans. :

Advantages of the version space method:

1. Can describe all the possible hypotheses in the language consistent with the data.
2. Fast (close to linear).

Disadvantages of the version space method:

1. Inconsistent data (noise) may cause the target concept to be pruned.
2. Learning disjunctive concepts is challenging

Q.27 Describe List-Then-Eliminate Algorithm.

Ans. : • List-Then-Eliminate algorithm initializes the version space to contain all hypotheses in H, then eliminates any hypothesis found inconsistent with any training example.

- The version space of candidate hypotheses thus shrinks as more examples are observed, until ideally just one hypothesis remains that is consistent with all the observed examples.
- If insufficient data is available to narrow the version space to a single hypothesis, then the algorithm can output the entire set of hypotheses consistent with the observed data.
- List-Then-Eliminate algorithm can be applied whenever the hypothesis space H is finite. It has many advantages, including the fact that it is guaranteed to output all hypotheses consistent with the training data.

- Unfortunately, it requires exhaustively enumerating all hypotheses in H - an unrealistic requirement for all but the most trivial hypothesis spa?ces

Q.28 Define candidate elimination algorithm.

Ans. : The candidate-Elimination algorithm computes the version space containing all (and only those) hypotheses from H that are consistent with an observed sequence of training examples

Q.29 Write algorithm for Candidate-Elimination.

Ans. :

Algorithm :

Given :

- A representation language.
- A set of positive and negative examples expressed in that language.

Compute : a concept description that is consistent with all the positive examples and none of the negative examples.

Method :

- Initialize G, the set of maximally general hypotheses, to contain one element : the null description (all features are variables).
- Initialize S, the set of maximally specific hypotheses, to contain one element : the first positive example.
- Accept a new training example.
- If the example is positive:

1. Generalize all the specific models to match the positive example, but ensure the following :
   - The new specific models involve minimal changes.
   - Each new specific model is a specialization of some general model.
   - No new specific model is a generalization of some other specific model.
2. Prune away all the general models that fail to match the positive example.
   - If the example is negative :

1. Specialize all general models to prevent match with the negative example, but ensure the following :
   - The new general models involve minimal changes.
   - Each new general model is a generalization of some specific model.
   - No new general model is a specialization of some other general model.
2. Prune away all the specific models that match the negative example.
   - If S and G are both singleton sets, then :
   - if they are identical, output their value and halt.
   - if they are different, the training cases were inconsistent. Output this result and halt.
   - else continue accepting new training examples.

The algorithm stops when :

1. It runs out of data.
2. The number of hypotheses remaining is :

0 - No consistent description for the data in the language.

1 - Answer (version space converges).

2^+ - All descriptions in the language are implicitly included.

Q.30 Explain candidate elimination algorithm with example.

Ans. : • The candidate-Elimination algorithm computes the version space containing all (and only those) hypotheses from H that are consistent with an observed sequence of training examples.

- Example : Learning the concept of "Japanese Economy Car"
- Features : Country of Origin, Manufacturer, Color, Decade, Type

Origin	Manufacturer	Color	Decade	Type	Example Type
Japan	Honda	Blue	1980	Economy	Positive
Japan	Toyota	Green	1970	Sports	Negative
Japan	Toyota	Blue	1990	Economy	Positive
USA	Chrysler	Red	1980	Economy	Negative
Japan	Honda	White	1980	Economy	Positive
Japan	Toyota	Green	1980	Economy	Positive
Japan	Honda	Red	1990	Economy	Negative

- **Positive Example 1 :** (Japan, Honda, Blue, 1980, Economy)
- Initialize G to a singleton set that includes everything.

$$G = \{ (?, ?, ?, ?, ?) \}$$

- Initialize S to a singleton set that includes the first positive example.

$$S = \{ (\text{Japan, Honda, Blue, 1980, Economy}) \}$$

- **Negative Example 2 :** (Japan, Toyota, Green, 1970, Sports)
- Specialize G to exclude the negative example.

$$G = \{ (?, \text{Honda}, ?, ?, ?), (?, ?, \text{Blue}, ?, ?), (?, ?, ?, 1980, ?), (?, ?, ?, ?, \text{Economy}) \}$$

$$S = \{ (\text{Japan, Honda, Blue, 1980, Economy}) \}$$

- **Positive Example 3** : (Japan, Toyota, Blue, 1990, Economy)
- Prune G to exclude descriptions inconsistent with the positive example.

$$G = \{ (?, ?, \text{Blue}, ?, ?), (?, ?, ?, ?, \text{Economy}) \}$$

- Generalize S to include the positive example :

$$S = \{ (\text{Japan}, ?, \text{Blue}, ?, \text{Economy}) \}$$

- **Negative Example :** (USA, Chrysler, Red, 1980, Economy)
- Specialize G to exclude the negative example (but stay consistent with S)

$$G = \{ (?, ?, \text{Blue}, ?, ?), (\text{Japan}, ?, ?, ?, \text{Economy}) \}$$

$$S = \{ (\text{Japan}, ?, \text{Blue}, ?, \text{Economy}) \}$$

Negative Example : (Japan, Honda, Red, 1990, Economy)

- Example is inconsistent with the version-space.

  G cannot be specialized.

  S cannot be generalized.

- The version space **collapses.**
- Conclusion : No conjunctive hypothesis is consistent with the data set

1.7 : Inductive Bias

Q.31 What is an inductive bias ?

Ans. :
- Consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and let Dc = {<x , c(x)>} be an arbitrary set of training examples of c.
- Let L(xi, Dc) denote the classification assigned to the instance xi by L after training on the data Dc.
- The inductive bias of L is any minimal set of assertions B such that for any target concept c and corresponding training examples Dc the following formula holds.

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \vdash L(x_i, D_c)]$$

Q.32 Explain fundamental property of Inductive Inference.

Ans. :
- A learner that makes no a priori assumptions regarding the identity of the target concept has no rational basis for classifying any unseen instances.
- Inductive Leap : A learner should be able to generalize training data using prior assumptions in order to classify unseen instances.
- The generalization is known as inductive leap and our prior assumptions are the inductive bias of the learner.
- Inductive Bias (prior assumptions) of Candidate-Elimination Algorithm is that the target concept can be represented by a conjunction of attribute values, the target concept is contained in the hypothesis space and training examples are correct.

Q.33 Explain with example inductive bias.

Ans. :
- The Candidate-Elimination algorithm will converge toward the true target concept provided it is given accurate training examples and provided its initial hypothesis space contains the target concept.
- What if the target concept is not contained in the hypothesis space ?
- Can we avoid this difficulty by using a hypothesis space that includes every possible hypothesis ?
- How does the size of this hypothesis space influence the ability of the algorithm to generalize to unobserved instances ?
- How does the size of the hypothesis space influence the number of training examples that must be observed ?
- In EnjoySport example, we restricted the hypothesis space to include only conjunctions of attribute values. Because of this restriction, the hypothesis space is unable to represent even simple disjunctive target concepts such as "Sky = Sunny or Sky = Cloudy."

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
1	Sunny	Warm	Normal	Strong	Cool	Change	YES
2	Cloudy	Warm	Normal	Strong	Cool	Change	YES
3	Rainy	Warm	Normal	Strong	Cool	Change	NO

- From first two examples : S2 : <?, Warm, Normal, Strong, Cool, Change>
- This is inconsistent with third examples, and there are no hypotheses consistent with these three examples PROBLEM : We have biased the learner to consider only conjunctive hypotheses. We require a more expressive hypothesis space.
- The obvious solution to the problem of assuring that the target concept is in the hypothesis space H is to provide a hypothesis space capable of representing every teachable concept.

Q.34 Which are the three learning algorithms from weakest to strongest bias ?

Ans. :
- ROTE-LEARNER : Learning corresponds simply to storing each observed training example in memory. Subsequent instances are classified by looking them up in memory. If the instance is found in memory, the stored classification is returned. Otherwise, the system refuses to classify the new instance. **Inductive Bias : No inductive bias**
- CANDIDATE-ELIMINATION : New instances are classified only in the case where all members of the current version space agree on the classification. Otherwise, the system refuses to classify the new instance. ***Inductive Bias : the target concept can be represented in its hypothesis space.***
- FIND-S : This algorithm, described earlier, finds the most specific hypothesis consistent with the training examples. It then uses this hypothesis to classify all subsequent instances. ***Inductive Bias: the target concept can be represented in its hypothesis space, and all instances are negative instances unless the opposite is entailed by its other knowledge.***

1.8 : Decision Tree Learning : Introduction and Representation

Q.35 What is decision tree ?

Ans. :
- Decision tree learning is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree.
- A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).
- A decision tree or a classification tree is a tree in which each internal node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature.

Q.36 What are the nodes of decision tree ?

Ans. :
- A decision tree has two kinds of nodes

1. Each leaf node has a class label, determined by majority vote of training examples reaching that leaf.
2. Each internal node is a question on features. It branches out according to the answers.

- Decision tree learning is a method for approximating discrete-valued target functions. The learned function is represented by a decision tree

Q.37 How to represents decision tree ?

Ans. : • Goal : Build a decision tree for classifying examples as positive or negative instances of a concept

- Supervised learning, batch processing of training examples, using a preference bias.
- A decision tree is a tree where
  a. Each non-leaf node has associated with it an attribute (feature)
  b. Each leaf node has associated with it a classification (+ or -)
  c. Each arc has associated with it one of the possible values of the attribute at the node from which the arc is directed.
- Internal node denotes a test on an attribute. Branch represents an outcome of the test. Leaf nodes represent class labels or class distribution.
- A decision tree is a flow-chart-like tree structure, where each node denotes a test on an attribute value, each branch represents an outcome of the test, and tree leaves represent classes or class distributions. Decision trees can easily be converted to classification rules.

Q.38 Write decision tree algorithm.

Ans. : • To generate decision tree from the training tuples of data partition D.

Input :

1. Data partition (D)
2. Attribute list
3. Attribute selection method

Algorithm :

1. Create a node (N)
2. If tuples in D are all of the same class then
3. Return node (N)as a leaf node labeled with the class C.
4. If attribute list is empty then return N as a leaf node labeled with the majority class in D
5. Apply attribute selection method(D, attribute list) to find the "best" splitting criterion;
6. Label node N with splitting criterion;
7. If splitting attribute is discrete-valued and multiway splits allowed
8. Then attribute list -> attribute list -> splitting attribute
9. For (each outcome j of splitting criterion)
10. Let D_j be the set of data tuples in D satisfying outcome j;
11. If D_j is empty then attach a leaf labeled with the majority class in D to node N;
12. Else attach the node returned by Generate decision tree (D_j , attribute list) to node N;
13. End of for loop
14. return N;

Q.39 Describe characteristics for appropriate problems for decision tree learning.

Ans. : • Decision tree learning is generally best suitec to problems with the following characteristics :

1. Instances are represented by attribute-value pairs. Fixed set of attributes, and the attributes take a small number of disjoint possible values.
2. The target function has discrete output values. Decision tree learning is appropriate for a boolean classification, but it easily extends to learning functions with more than two possible output values.
3. Disjunctive descriptions may be required. Decision trees naturally represent disjunctive expressions.
4. The training data may contain errors. Decision tree learning methods are robust to errors, both errors in classifications of the training examples and errors in the attribute values that describe these examples.
5. The training data may contain missing attribute values. Decision tree methods can be used even when some training examples have unknown values.
6. Decision tree learning has been applied to problems such as learning to classify

Q.40 Define the following :
a. Input variable b. Leaf node c. Internal nodes d. Depth

Ans. :

a. **Input variable :** Each member of the set { x_1, x_2, ... x_n} is called an input variable.

b. **Leaf node :** A node without further branches is called a leaf node. The leaf nodes return class

labels and, in some implementations, they return the probability scores.

c. **Internal nodes** are the decision or test points. Each internal node refers to an input variable or an attribute. The top internal node is called the root.

d. **Depth** : The depth of a node is the minimum number of steps required to reach the node from the root.

Q.41 List the advantages and disadvantages of decision tree.

Ans. : Advantages :

1. Rules are simple and easy to understand.
2. Decision trees can handle both nominal and numerical attributes.
3. Decision trees are capable of handling datasets that may have errors.
4. Decision trees are capable of handling datasets that may have missing values.
5. Decision trees are considered to be a nonparametric method.
6. Decision trees are self-explanatory.

Disadvantages :

1. Most of the algorithms require that the target attribute will have only discrete values.
2. Some problem are difficult to solve like XOR.
3. Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
4. Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

Q.42 How to evaluate a decision tree ?

Ans. : • A decision tree can help you make tough choices between different paths and outcomes, but only if you evaluate the model correctly.

- Decision trees are graphic models of possible decisions and all related possible outcomes in a tree form, with the outcomes shown as "branches" off each choice. You can use a decision tree to help you make all kinds of business decisions, including new product development, new marketing strategies and workforce changes.
- Decision tree is evaluated as follows :
  1. First, evaluate whether the splits of the tree make sense. Conduct sanity checks by validating the decision rules with domain experts, and determine if the decision rules are sound.
- Next, look at the depth and nodes of the tree. Having too many layers and obtaining nodes with few members might be signs of overfitting.
- In overfitting, the model fits the training set well, but it performs poorly on the new samples in the testing set.
- Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship.
- Overfitting is when a classifier fits the training data too tightly. Such a classifier works well on the training data but not on independent test data. It is a general problem that plagues all machine learning methods.
- Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.
- To prevent over-fitting we have several options :
  1. Restrict the number of adjustable parameters the network has - e.g. by reducing the number of hidden units, or by forcing connections to share the same weight values.
  2. Stop the training early, before it has had time to learn the training data too well.
  3. Add some form of regularization term to the error/cost function to encourage smoother network mappings.
  4. Add noise to the training patterns to smear out the data points.

1.9 : Basic Decision Tree Learning Algorithm

Q.43 Define information gain.

Ans. : • Entropy measures the impurity of a collection. Information Gain is defined in terms of Entropy.

- Information gain tells us how important a given attribute of the feature vectors is.

- Information gain of attribute A is the reduction in entropy caused by partitioning the set of examples S.

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

where Values (A) is the set of all possible values for attribute A and S_v is the subset of S for which attribute A has value v

Q.44 What is the role of information gain in decision tree learning ?

[JNTU : Dec.-16, Marks-3]

Ans. : • Information gain measures how well a given attribute separates the training examples according to their target classification

- ID3 uses this information gain measure to select among the candidate attributes at each step while growing the tree.
- Entropy is used for measuring information gain.
- Putting together a decision tree is all a matter of choosing which attribute to test at each node in the tree.
- We shall define a measure called information gain which will be used to decide which attribute to test at each node.
- Information gain is itself calculated using a measure called entropy, which we first define for the case of a binary decision problem and then define for the general case

Q.45 Explain Gini Index and Entropy of dicision tree algorithm.

Ans. : • One of the decision tree algorithms is CART (Classification and Regression Tree).

- Classification Tree : When decision or target variable is categorical, the decision tree is classification decision tree.
- Regression Tree : When the decision or target variable is continuous variable, the decision tree is called regression decision tree.
- CART algorithm can be used for building both Classification and Regression Decision Trees. The impurity measure used in building decision tree in CART is Gini Index. The decision tree built by CART algorithm is always a binary decision tree.
- Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.
- Gini index, entropy and twoing rule are some of the frequently used impurity measures.
- Gini Index for a given node t :

$$\text{GINI}(t) = \sum_j p(j|t)(1-p(j|t)) = \sum_j p(j|t)^2$$

Maximum of $1-1/n_c$ (number of classes) when records are equally distributed among all classes = maximal impurity.

- Minimum of 0 when all records belong to one class = complete purity.
- Entropy at a given node by :

$$\text{Entropy}(t) = \sum_j p(j|t) \log p(j|t)$$

- Maximum (log n_c) when records are equally distributed among all classes(maximal impurity).
- Minimum (0.0) when all records belongs to one class (maximal purity).
- Entropy is the only function that satisfies all of the following three properties
  1. When node is pure, measure should be zero
  2. When impurity is maximal (i.e. all classes equally likely), measure should be maximal
  3. Measure should obey multistage property
- When a node p is split into k partitions (children), the quality of the split is computed as a weighted sum :

$$\text{GINI}_{\text{split}} = \sum_{i=1}^{k} \frac{n_i}{n} \text{GINI}(i) = \sum_j p(j|t)^2$$

where n_i = number of records at child i, and n = number of records at node p.

- A problem with all impurity measures is that they depend only on the number of (training) patterns of different classes on either side of the hyperplane. Thus, if we change the class regions without changing the effective areas of class regions on

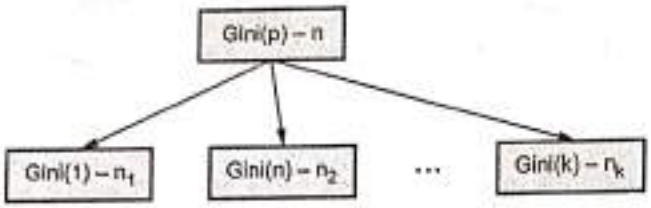


Fig. Q.45.1

either side of a hyperplane, the impurity measure of the hyperplane will not change.

- Thus the impurity measures do not really capture the geometric structure of class distributions. Also, all the algorithms need to optimize on some average of impurity of the child nodes and often it is not clear what kind of average is proper.

Q.46 Which attribute is best ? How to select best attributes ?

Ans. : • We would like to select the attribute that is most useful for classifying examples.

- Information gain measures how well a given attribute separates the training examples according to their target classification.
- ID3 uses this information gain measure to select among the candidate attributes at each step while growing the tree.
- In order to define information gain precisely, we use a measure commonly used in information theory, called entropy
- Entropy characterizes the impurity of an arbitrary collection of examples.
- Putting together a decision tree is all a matter of choosing which attribute to test at each node in the tree.
- We shall define a measure called information gain which will be used to decide which attribute to test at each node.
- Information gain is itself calculated using a measure called entropy, which we first define for the case of a binary decision problem and then define for the general case.
- Given a binary categorization, C, and a set of examples, S, for which the proportion of examples categorized as positive by C is p+ and the proportion of examples categorized as negative by C is p-, then the entropy of S is :

$$\text{Entropy}(S) = -p_+ \log_2 (p_+) - p - \log_2 (p_-)$$

Where

S is a sample of training examples

p+ is the proportion of positive examples

p- is the proportion of negative examples

- Imagine having a set of boxes with some balls in. If all the balls were in a single box, then this would be nicely ordered, and it would be extremely easy to find a particular ball.
- If, however, the balls were distributed amongst the boxes, this would not be so nicely ordered, and it might take quite a while to find a particular ball.
- If we were going to define a measure based on this notion of purity, we would want to be able to calculate a value for each box based on the number of balls in it, then take the sum of these as the overall measure.
- We would want to reward two situations: nearly empty boxes, and boxes with nearly all the balls in. This is the basis for the general entropy measure, which is defined as follows.
- Given an arbitrary categorization, C into categories $c_1, \ldots, c_n$ and a set of examples, S, for which the proportion of examples in c_i is p_i, then the entropy of S is :

$$\text{Entropy}(S) = \sum_{i=1}^{n} -p_i \log_2 (p_i)$$

Q.47 Explain ID3 algorithm.

Ans. : • The calculation for information gain is the most difficult part of this algorithm.

- ID3 performs a search whereby the search states are decision trees and the operator involves adding a node to an existing tree. It uses information gain to measure the attribute to put in each node, and performs a greedy search using this measure of worth.

- The algorithm goes as follows : Given a set of examples (S), categorised in categories c_i, then :

1. Choose the root node to be the attribute, A, which scores the highest for information gain relative to S.
2. For each value v that A can possibly take, draw a branch from the node.
3. For each branch from A corresponding to value v, calculate S_v. Then :
   i. If S_v is empty, choose the category c default which contains the most examples from S, and put this as the leaf node category which ends that branch.
   ii. If S_v contains only examples from a category c, then put c as the leaf node category which ends that branch.
   iii. Otherwise, remove A from the set of attributes which can be put into nodes. Then put a new node in the decision tree, where the new attribute being tested in the node is the one which scores highest for information gain relative to S_v.

The following diagram should explain the ID3 algorithm further.

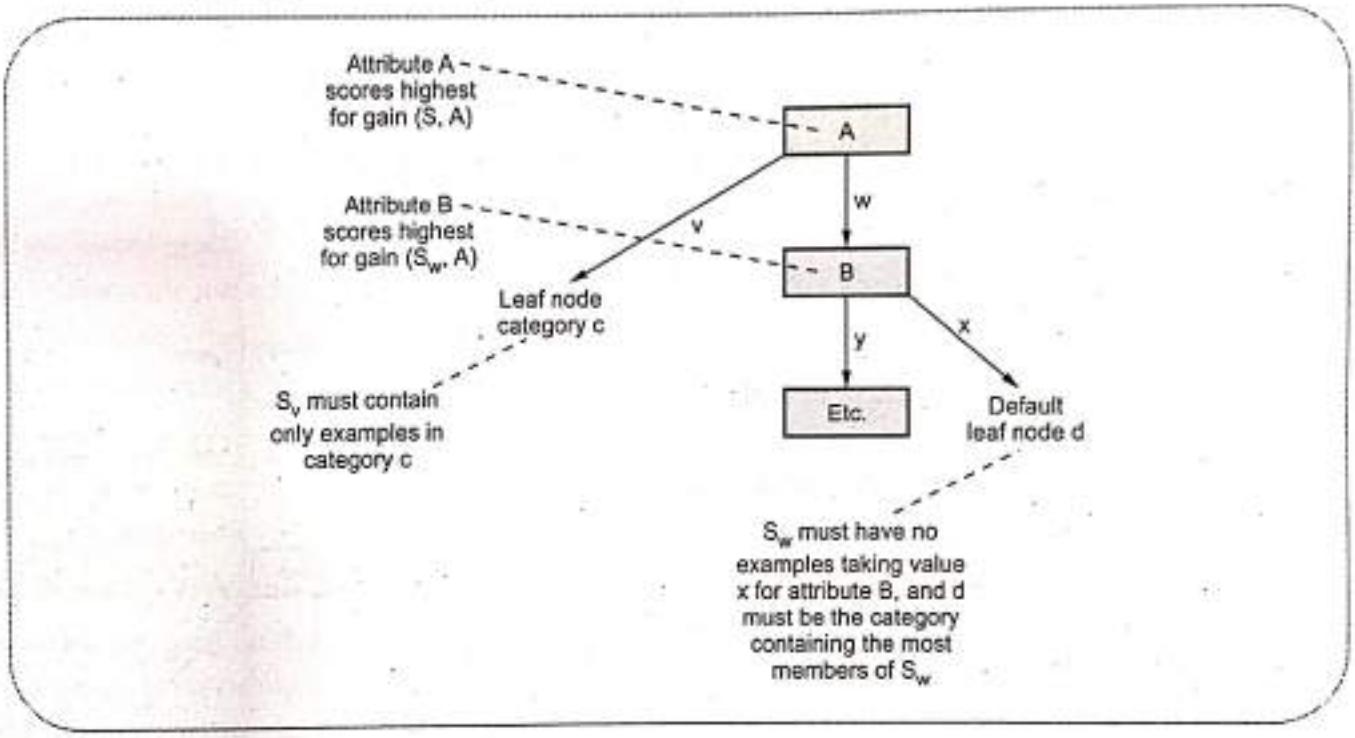


Fig. Q.47.1

Q.48 Suppose we want to train a decision tree using the following instances :

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W3	Windy	Yes	Rich	Cinema
W4	Rainy	Yes	Poor	Cinema
W5	Rainy	No	Rich	Stay in
W6	Rainy	Yes	Poor	Cinema

W7	Windy	No	Poor	Cinema
W8	Windy	No	Rich	Shopping
W9	Windy	Yes	Rich	Cinema
W10	Sunny	No	Rich	Tennis

Ans. :

$$\text{Entropy(S)} = -p_{\text{cinema}}\log_2(p_{\text{cinema}}) - p_{\text{tennis}}\log_2(p_{\text{tennis}}) - p_{\text{shopping}}\log_2(p_{\text{shopping}}) - p_{\text{stay\_in}}\log_2(p_{\text{stay\_in}})$$

$$= -(6/10) * \log_2(6/10) - (2/10) * \log_2(2/10) - (1/10) * \log_2(1/10) - (1/10) * \log_2(1/10)$$

$$= -(6/10) * -0.737 - (2/10) * -2.322 - (1/10) * -3.322 - (1/10) * -3.322$$

$$= 0.4422 + 0.4644 + 0.3322 + 0.3322 = 1.571$$

and we need to determine the best of :

$$\text{Gain(S, weather)} = 1.571 - (|S_{\text{sun}}|/10) * \text{Entropy}(S_{\text{sun}}) - (|S_{\text{wind}}|/10) * \text{Entropy}(S_{\text{wind}}) - (|S_{\text{rain}}|/10) * \text{Entropy}(S_{\text{rain}})$$

$$= 1.571 - (0.3) * \text{Entropy}(S_{\text{sun}}) - (0.4) * \text{Entropy}(S_{\text{wind}}) - (0.3) * \text{Entropy}(S_{\text{rain}})$$

$$1.571 - (0.3) * (0.918) - (0.4) * (0.81125) - (0.3) * (0.918) = 0.70$$

$$\text{Gain(S, parents)} = 1.571 - (|S_{\text{yes}}|/10) * \text{Entropy}(S_{\text{yes}}) - (|S_{\text{no}}|/10) * \text{Entropy}(S_{\text{no}})$$

$$= 1.571 - (0.5) * 0 - (0.5) * 1.922 = 1.571 - 0.961 = 0.61$$

$$\text{Gain(S, money)} = 1.571 - (|S_{\text{rich}}|/10) * \text{Entropy}(S_{\text{rich}}) - (|S_{\text{poor}}|/10) * \text{Entropy}(S_{\text{poor}})$$

$$1.571 - (0.7) * (1.842) - (0.3) * 0 = 1.571 - 1.2894 = 0.2816$$

- This means that the first node in the decision tree will be the weather attribute. From the weather node, we draw a branch for the values that weather can take: sunny, windy and rainy :
- Now we look at the first branch. S_{sunny} = {W1, W2, W10}. This is not empty, so we do not put a default categorization leaf node here.
- The categorizations of W1, W2 and W10 are Cinema, Tennis respectively. As these are not all the same, we cannot put a categorization leaf node here. Hence we put an attribute node here, which we will leave blank for the time being.
- Looking at the second branch, S_{windy} = {W3, W7, W8, W9}. Again, this is not empty, and they do not all belong to the same class, so we put an attribute node here, left blank for now. The same situation happens with the third branch, hence our amended tree looks like this :

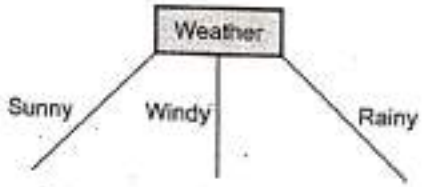


Fig. Q.48.1

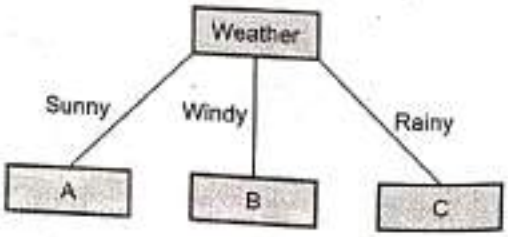


Fig. Q.48.2

- In effect, we are interested only in this part of the table :

Weekend (Example)	Weather	Parents	Money	Decision (Category)
W1	Sunny	Yes	Rich	Cinema
W2	Sunny	No	Rich	Tennis
W10	Sunny	No	Rich	Tennis

Hence we can calculate :

$$\text{Gain}(S_{sunny}, \text{parents}) = 0.918 - (|S_{yes}|/|S|) * \text{Entropy}(S_{yes}) - (|S_{no}|/|S|) * \text{Entropy}(S_{no})$$

$$= 0.918 - (1/3) * 0 - (2/3) * 0 = 0.918$$

$$\text{Gain}(S_{sunny}, \text{money}) = 0918 - (|S_{rich}|/|S|) * \text{Entropy}(S_{rich}) - (|S_{poor}|/|S|) * \text{Entropy}(S_{poor})$$

$$= 0.918 - (3/3) * 0918 - (0/3) * 0 = 0.918 - (3/3) * 0.918 - (0/3) * 0 = 0.918 - 0.918 = 0$$

1.10 : Hypothesis Space Search in Decision Tree Learning

Q.49 Discuss hypothesis space search in decision tree learning.

Ans. : • The hypothesis space searched by ID3 is the set of possible decision trees. ID3 performs a simple-to complex, hill-climbing search through this hypothesis space.

- Begins with the empty tree, then considers progressively more elaborate hypotheses in search of a decision tree that correctly classifies the training data.
- The information gain measure guides the hill-climbing search.
- **Hypothesis Space :** Set of possible decision trees
- **Search Method :** Simple-to-Complex Hill-Climbing Search (only a single current hypothesis is maintained (from candidate-elimination method)). No Backtracking!!!
- **Evaluation Function :** Information Gain Measure
- **Batch Learning :** ID3 uses all training examples at each step to make statistically-based decisions (from candidate-elimination method which makes decisions incrementally). The search is less sensitive to errors in individual training examples.
- By viewing ID3 in terms of its search space and search strategy, following are the capabilities and limitations :
  1. ID3 hypothesis space of all decision trees is a complete space of finite discrete-valued functions, relative to the available attributes.
  2. It maintains only a single current hypothesis as it searches through the space of decision trees.
  3. ID3 in its pure form performs no backtracking in its search
  4. ID3 uses all training examples at each step in the search to make statistically based decisions regarding how to refine its current hypothesis

1.11 : Inductive Bias in Decision Tree Learning

Q.50 Compare ID3 with Candidate-Elimination algorithm.

Ans. :

ID3 Algorithm	Candidate-Elimination algorithm
ID3 searches a complete hypothesis space	The version space CANDIDATE-ELIMINAATION searches an incomplete hypothesis space
Its hypothesis space introduces no additional bias	Its search strategy introduces no additional bias.
Its inductive bias is solely a consequence of the ordering of hypotheses by its search strategy.	Its inductive bias is solely a consequence of the expressive power of its hypothesis representation
It searches incompletely	It searches space completely

Q.51 Why Prefer Short Hypotheses ?

Ans. : Argument in favor:

1. Fewer short hypotheses than long hypotheses
2. A short hypothesis that fits the data is unlikely to be a coincidence
3. A long hypothesis that fits the data might be a coincidence

Argument opposed :

1. There are many ways to define small sets of hypotheses
2. What is so special about small sets based on size of hypothesis
   - **OCCAM'S RAZOR** : Prefer the simplest hypothesis that fits the data.
   - **Occam's razor** was shown experimentally to be a successful strategy.
   - The term Occam's razor refers to the philosophical idea or scientific principle that of any given set of explanations for an event occurring, it is most likely that the simplest one is the correct one.
   - Occam's razor does not seek to offer complete and absolute proof, but to find the simplest probable answer to a question of why an event happened

Q.52 State Occam's razor principle. [JNTU : Dec.-16, Marks 2]

Ans. :
- Prefer the simplest hypothesis that fits the data
- Occam's razor will produce two different hypotheses from the same training examples when it is applied by two learners that perceive these examples in terms of different internal representations

1.12 : Issues in Decision Tree Learning

Q.53 Define pre pruning and post pruning.

Ans. : • In prepruning, a tree is "pruned" by halting its construction early. Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset tuples or the probability distribution of those tuples.

- In the postpruning, it removes subtrees from a "fully grown" tree. A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labeled with the most frequent class among the subtree being replaced.

Q.54 Why tree pruning useful in decision tree induction ?

Ans. : When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of **overfitting** the data. Such methods typically use statistical measures to remove the least reliable branches.

Q.55 What is tree pruning ?

Ans. : Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data

Q.56 Explain overfitting. What are the reason for overfitting ?

Ans. : • Training error can be reduced by making the hypothesis more sensitive to training data, but this may lead to overfitting and poor generalization.

- Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship.
- Overfitting is when a classifier fits the training data too tightly. Such a classifier works well on the training data but not on independent test data. It is a general problem that plagues all machine learning methods.
- Because of overfitting, low error on training data and high error on test data.
- Overfitting occurs when a model begins to memorize training data rather than learning to generalize from trend.
- The more difficult a criterion is to predict, the more noise exists in past information that need to be ignored. The problem is determining which part to ignore.
- Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.
- We can determine whether a predictive model is underfitting or overfitting the training data by looking at the prediction error on the training data and the evaluation data.
- Fig. Q.56.1 shows overfiting.

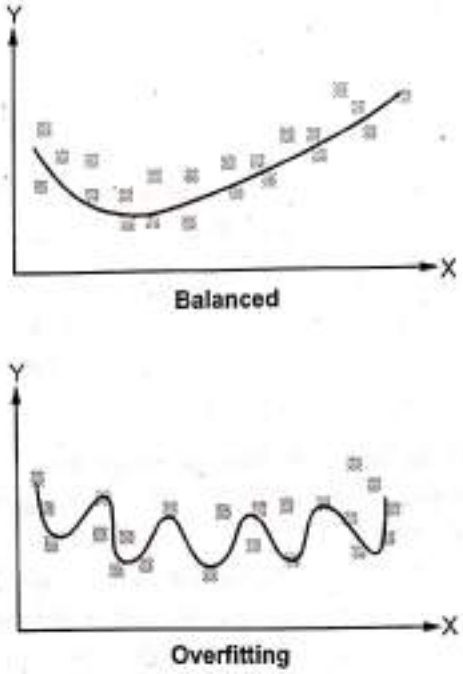


Fig. Q.56.1 Overfitting

- Reasons for overfitting
  1. Noisy data 2. Training set is too small
  3. Large number of features
- Method to avoid overfittings :
  1. Limit the number of hidden nodes.
  2. Stop training early to avoid a perfect explanation of the training set, and
  3. Apply weight decay to limit the size of the weights, and thus of the function class implemented by the network.

Q.57 Why is tree pruning useful in decision tree induction ? What is a drawback of using a separate set of tuples to evaluate pruning ?

Ans. : • The decision tree built may overfit the training data. There could be too many branches, some of which may reflect anomalies in the training data due to noise or outliers.

- Tree pruning addresses this issue of overfitting the data by removing the least reliable branches.
- This generally results in a more compact and reliable decision tree that is faster and more accurate in its classification of data.
- The drawback of using a separate set of tuples to evaluate pruning is that it may not be representative of the training tuples used to create the original decision tree.
- If the separate set of tuples are skewed, then using them to evaluate the pruned tree would not be a good indicator of the pruned tree's classification accuracy.
- Furthermore, using a separate set of tuples to evaluate pruning means there are less tuples to use for creation and testing of the tree.
- While this is considered a drawback in machine learning, it may not be so in data mining due to the availability of larger data sets.

Q.58 What is decision tree pruning? Explain error based pruning and reduced error pruning method for tree pruning.

Ans. : • Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.

- Pruning reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting
- Pruning means to change the model by deleting the child nodes of a branch node. The pruned node is regarded as a leaf node. Leaf nodes cannot be pruned.
- A decision tree consists of a root node, several branch nodes, and several leaf nodes.

1. The root node represents the top of the tree. It does not have a parent node, however, it has different child nodes.
2. Branch nodes are in the middle of the tree. A branch node has a parent node and several child nodes.
3. Leaf nodes represent the bottom of the tree. A leaf node has a parent node. It does not have child nodes.

Error based pruning :

- Error based pruning can be used to prune a decision tree and it does not require the use of validation data.
- Error-based pruning considers the E errors among the N training examples at a leaf of the tree to give an estimate of the error probability for that node.
- The assumption is that these are E events in N independent trials which is, of course, not perfectly true.
- Using the binomial theorem, condence limits can be calculated for the probability of error for a given condence level
- EBP also performs subtree raising. In the case of subtree raising, an internal node can be replaced by the subtree of one of its children rather than a leaf.
- EBP will estimate that more errors are made than actually observed on the validation set for certainty factors lower than 100. Hence, if we ignore subtree raising, then EBP applied to a validation set will tend to prune more than reduced error pruning (REP) for all CF < 100 and will certainly prune more of the tree for smaller CFs.

Reduced Error Pruning

- It is simplest and most understandable method in decision tree pruning.
- This method considers each of the decision nodes in the tree to be candidates for pruning, consist of removing the subtree rooted at that node, making it a leaf node.
- The available data is divided into three parts : the training examples, the validation examples used for pruning the tree, and a set of test examples used to provide an unbiased estimate of accuracy over future unseen examples.
- If the error rate of the new tree would be equal to or smaller than that of the original tree and that subtree contains no subtree with the same property, then subtree is replaced by leaf node, means pruning is done.

- Otherwise don't prune it. The advantage of this method is its linear computational complexity.
- When the test set is much smaller than the training set, this method may lead to over pruning.

Q.59 What is RULE POST-PRUNING ?

Ans. : • It is method for finding high accuracy hypotheses.

- Rule post-pruning involves the following steps :
  1. Infer decision tree from training set
  2. Convert tree to rules - one rule per branch
  3. Prune each rule by removing preconditions that result in improved estimated accuracy
  4. Sort the pruned rules by their estimated accuracy and consider them in this sequence when classifying unseen instances

Q.60 Why convert the decision tree to rules before pruning ?

Ans. :

- Converting to rules allows distinguishing among the different contexts in which a decision node is used.
- Converting to rules removes the distinction between attribute tests that occur near the root of the tree and those that occur near the leaves.
- Converting to rules improves readability. Rules are often easier for to understand

Q.61 Explain alternative measures for selecting attributes.

Ans. : • One of the decision tree algorithms is CART (Classification and Regression Tree).

- Classification Tree : When decision or target variable is categorical, the decision tree is classification decision tree.
- Regression Tree : When the decision or target variable is continuous variable, the decision tree is called regression decision tree.
- CART algorithm can be used for building both Classification and Regression Decision Trees. The impurity measure used in building decision tree in CART is Gini Index. The decision tree built by CART algorithm is always a binary decision tree.
- Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.
- Gini index, entropy and twoing rule are some of the frequently used impurity measures.
- Gini Index for a given node t :

$$GINI(t) = \sum_j p(j|t)(1-p(j|t)) = \sum_j p(j|t)^2$$

Maximum of $1-1/n_c$ (number of classes) when records are equally distributed among all classes = maximal impurity.

- Minimum of 0 when all records belong to one class = complete purity.
- Entropy at a given node by :

$$\text{Entropy (t)} = \sum_j p(j|t) \log p(j|t)$$

- Maximum (log n_c) when records are equally distributed among all classes(maximal impurity).
- Minimum (0.0) when all records belongs to one class (maximal purity).
- Entropy is the only function that satisfies all of the following three properties
  1. When node is pure, measure should be zero
  2. When impurity is maximal (i.e. all classes equally likely), measure should be maximal
  3. Measure should obey multistage property
- When a node p is split into k partitions (children), the quality of the split is computed as a weighted sum :

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i) = \sum_j p(j|t)^2$$

where n_i = number of records at child i, and n = number of records at node p.

- A problem with all impurity measures is that they depend only on the number of (training) patterns of different classes on either side of the hyperplane. Thus, if we change the class regions without changing the effective areas of class regions on either side of a hyperplane, the impurity measure of the hyperplane will not change.

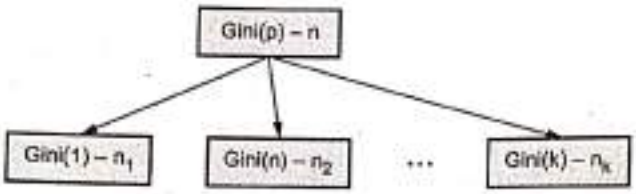


Fig. Q.61.1

- Thus the impurity measures do not really capture the geometric structure of class distributions. Also, all the algorithms need to optimize on some average of impurity of the child nodes and often it is not clear what kind of average is proper.

Fill in the Blanks for Mid Term Exam

Q.1 Machine Learning is a sub-field of _____ which concerns with developing computational theories of learning and building learning machines.

Q.2 _____ can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation.

Q.3 The key property of the _____ algorithm is that for hypothesis spaces described by conjunctions of attribute constraints.

Q.4 The _____ algorithm finds all describable hypotheses that are consistent with the observed training examples

Q.5 A hypothesis h is _____ with a set of training examples D if and only if h(x) = c(x) for each example (x, c(x)) in D.

Q.6 The LIST-THEN-ELIMINATE first initializes the _____ to contain all hypotheses in H, then eliminates any hypothesis found inconsistent with any training example

Q.7 Decision tree induction is the learning of decision trees from _____ training tuples.

Q.8 A _____ is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node holds a class label.

Q.9 _____ uses information gain as its attribute selection measure

Q.10 CART stands for _____.

Q.11 A _____ set of class-labeled tuples is used to estimate cost complexity

Q.12 Theoretical results have been developed that characterize the fundamentalrelationship among the number of _____ examples observed. [JNTU : August-16]

Q.13 All of the most successful _____ employ machine learning in some form. [JNTU : August-16]

Q.14 _____ is a significant practical difficulty for decision tree learning and many other learning methods. [JNTU : August-16]

Q.15 In learning to play checkers, the system might learn from_____training examples consisting of individual checkers board states and the correct move for each. [JNTU : Feb.-17]

Q.16 Learning algorithms to acquire only some approximation to the target function, and for this reason the process of learning the target function is often called_____. [JNTU : Feb.-17]

Multiple Choice Question for Mid Term Exam

Q.1 _____ finds the most specific hypothesis consistent with the training examples.

a FIND-S b ROTE-LEARN
c CANDIDATE-ELIMINATION
d All of these

Q.2 The CANDIDATE-ELIMINATION has a _____, that the target concept can be represented in its hypothesis space.

a no inductive bias
b inductive bias
c stronger inductive bias
d none of these

UNIT - II

2 Artificial Neural Network

2.1 : Introduction

Q.1 What is artificial neural network ?

Ans. : An (artificial) neural network consists of units, connections and weights. Inputs and outputs are numeric.

Q.2 Define the term neural network.

Ans : Neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

Q.3 Where are neural networks applicable?

Ans :

a. **In signature analysis :** as a mechanism for comparing signatures made with those stored.

b. **In process control :** there are clearly applications to be made here, most processes cannot be determined as computable algorithms.

c. **In monitoring :** networks have been used to monitor the state of aircraft engines.

Q.4 List advantages of Neural Networks

Ans : The advantages of neural networks are due to its adaptive and generalization ability.

a) Neural networks are adaptive methods that can learn without any prior assumption of the underlying data.

b) Neural network, namely the feed forward multilayer perception and radial basis function network have been proven to be universal functional approximations.

c) Neural networks are non-linear model with good generalization ability.

Q.5 Define structure and function of single neuron.

Ans. : • Artificial neural systems are inspired by biological neural systems. The elementary building block of biological neural systems is the neuron.

• Fig. Q.5.1 shows biological neural systems.

• The single cell neuron consists of the cell body or soma, the dendrites and the axon. The dendrites receive signals from the axons of other neurons.

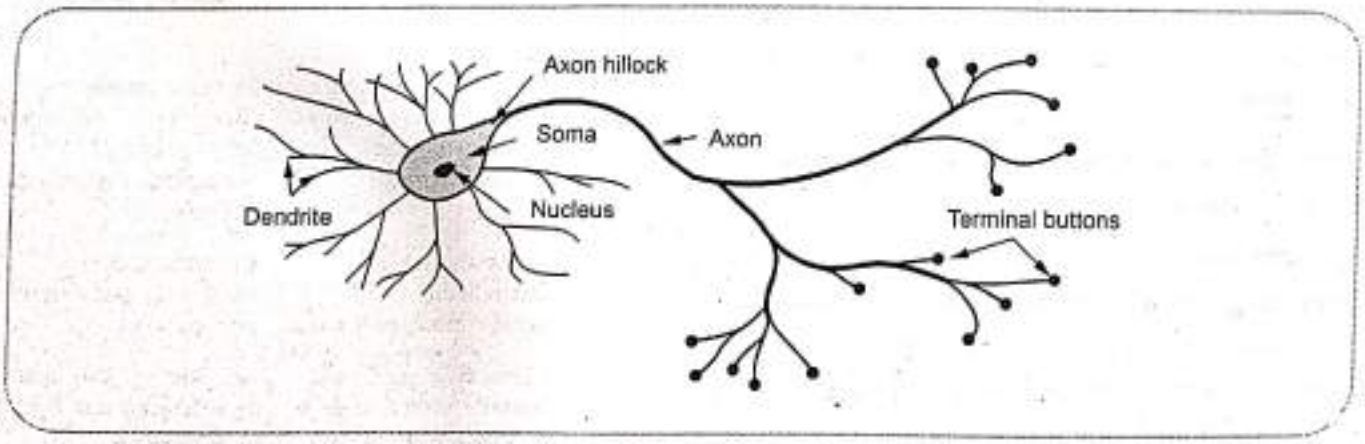


Fig. Q.5.1 Schematic of biological neuron

- The small space between the axon of one neuron and the dendrite of another is the synapse. The afferent dendrites conduct impulses toward the soma. The efferent axon conducts impulses away from the soma.
- Basic Components of Biological Neurons

1. The majority of neurons encode their activations or outputs as a series of brief electrical pulses.
2. The neuron's cell body (soma) processes the incoming activations and converts them into output activations.
3. The neuron's nucleus contains the genetic material in the form of DNA. This exists in most types of cells, not just neurons.
4. Dendrites are fibres which emanate from the cell body and provide the receptive zones that receive activation from other neurons.
5. Axons are fibres acting as transmission lines that send activation to other neurons.
6. The junctions that allow signal transmission between the axons and dendrites are called synapses. The process of transmission is by diffusion of chemicals called neuro transmitters across the synaptic cleft.

Q.6 What are the characteristics and application of ANN ? **[RGPV : Dec.-16, Marks 2]**

Ans. : Characteristics of Artificial Neural Networks

1. Large number of very simple processing neuron-like processing elements.
2. Large number of weighted connections between the elements.
3. Distributed representation of knowledge over the connections.
4. Knowledge is acquired by network through a learning process.

Application of ANN :

1. Controlling the movements of a robot based on self-perception and other information;
2. Deciding the category of potential food items in an artificial world;
3. Recognizing a visual object;
4. Predicting where a moving object goes, when a robot wants to catch it.

Q.7 List out the strength and weakness of artificial neural network.

Ans. : Strength :

1. The greatest power of Neural Networks is that it is endowed with a finite number of hidden units, can yet approximate any continuous function to any desired degree of accuracy. This has been commonly referred to as the property of universal approximate.
2. No prior knowledge of the data generating process is needed for implementing NN.
3. Problem of model misspecification does not occur.
4. In case of NN since no specifications are used as the network merely learns the hidden relationship in the data.
5. **Adaptive learning :** An ability to learn how to do tasks based on the data given for training or initial experience.
6. **Self-Organisation :** An ANN can create its own organisation or representation of the information it receives during learning time.

Weakness :

1. The addition of too many hidden units incites the problem of over fitting the data.
2. The construction of the NN model can be a time consuming process.

Q.8 Difference between digital computer and neural network.

Ans. :

Sr. No.	Digital computer	Neural network
1.	Deductive reasoning : We apply known rules to input data to produce output.	Inductive reasoning : Given input and output data (training examples), we construct the rules.
2.	Computation is centralized, synchronous and serial.	Computation is collective, asynchronous and parallel.
3.	Memory is packetted, literally stored and location addressable.	Memory is distributed, internalized and content addressable.

4.	Not fault tolerant. One transistor goes and it no longer works.	Fault tolerant, redundancy and sharing of responsibilities.
5.	Fast. Measured in millionths of a second.	Slow. Measured in thousandths of a second.
6.	Exact.	Inexact.
7.	Static connectivity.	Dynamic connectivity.
8.	Applicable if well defined rules with precise input data.	Applicable if rules are unknown or complicated or if data is noisy or partial.

Q.9 Explain the neural network architectures.

[RGPV : June-14, Marks 7]

Ans. : • The architecture of the neural network refers to the arrangement of the connection between neurons, processing element, number of layers, and the flow of signal in the neural network.

- There are mainly two category of neural network architecture :
  a. Feed-forward
  b. Feedback (recurrent) neural networks.

1. Architecture and Learning Rule

- In late 1950s, Frank Rosenblatt introduced a network composed of the units that were enhanced version of McCulloch-Pitts Threshold Logic Unit (TLU) model.
- Rosenblatt's model of neuron, a perceptron, was the result of merger between two concepts from the 1940s, McCulloch-Pitts model of an artificial neuron and Hebbian learning rule of adjusting weights.
- In addition to the variable weight values, the perceptron model added an extra input that represents bias. Thus, the modified equation is now as follows :

$$\text{Sum} = \sum_{i=1}^{N} I_i \; W_i + b$$

where **b** represents the bias value.

- Fig. Q.9.1 shows a typical perception setup for pattern recognition applications, in which visual patterns are represented as matrices of elements between 0 and 1.

1. First layer act as a set of feature detectors that are hardwired to the input signals to detect specific features.
2. Second layer i.e. output layer takes the outputs of the feature detectors in the first layer and classifies the given input pattern.

- Learning is initiated by making adjustments to the relevant connection strengths and a threshold value θ.
- Here we consider only two class problem. Here output layer usually has only a single node. For an n-class problem (n > 3), the output layer usually has n-nodes, each corresponding to a class and the output node with the largest value indicates which class the input vector belongs to.

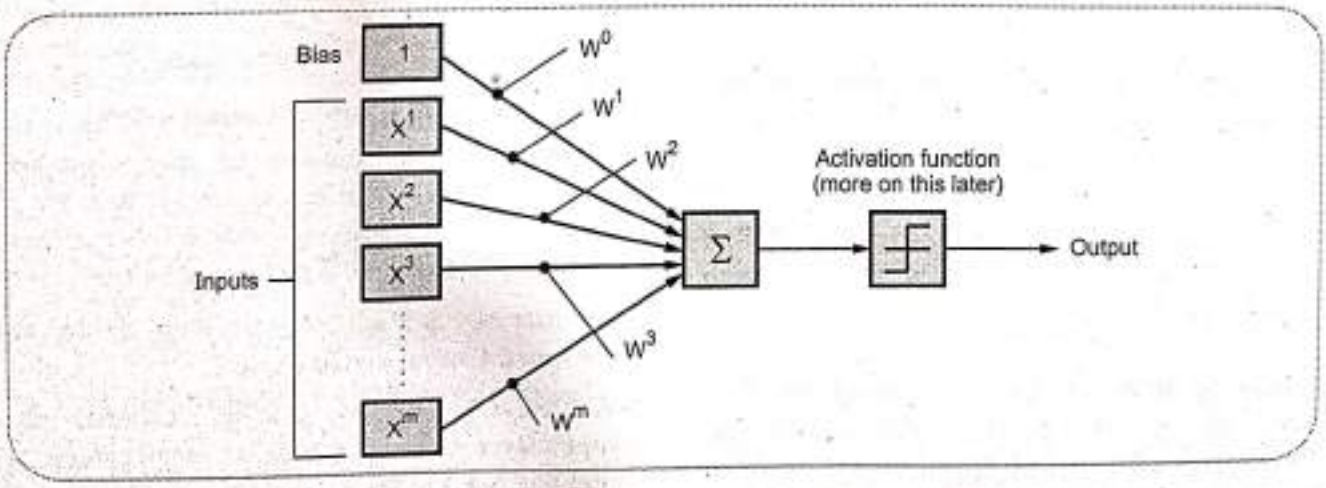


Fig. Q.9.1

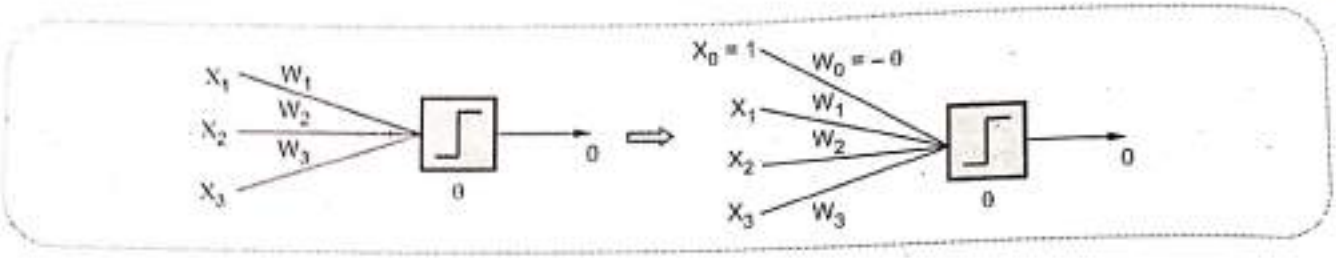


Fig. Q.9.2 Bias term w_0

- In the first stage, the linear combination of inputs is calculated. Each value of input array is associated with its weight value, which is normally between 0 and 1. Also, the summation function often takes an extra input value Theta with weight value of 1 to represent threshold or bias of a neuron.
  a. The term x_i is referred to as **active or excitatory** if its value is 1.
  b. If the value is 0 then it is **inactive.**
  c. If the value is –1 then it is **inhibitory.**
- The output unit is a linear threshold element with a threshold value θ :

$$0 = f\left(\sum_{i=1}^{n} w_i\ x_i - \theta\right)$$

$$= f\left(\sum_{i=1}^{n} w_i\ x_i + w_0\right), w_0 = -\theta$$

$$= f\left(\sum_{i=1}^{n} w_i\ x_i\right),\ x_0 = 1$$

where w_i is a modifiable weight associated with an incoming signal x_i.

- Fig. Q.9.2 shows the bias term w_0.
- The function $y = f(x)$ describes relationship, an input-output mapping from x to y.
- The equation (1), the f(.) is the **activation function** of the perceptron and it is typically either a **signum function** sgn(x) or **step function** step(x) :

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{otherwise} \end{cases}$$

$$\text{step}(x) = \begin{cases} 1 & \text{if } x > 0, \\ 0 & \text{otherwise} \end{cases} \quad \ldots (1)$$

- The sum-of-product value is then passed into the second stage to perform the activation function which generates the output from the neuron. The activation function "squashes" the amplitude of the output in the range of [0, 1] or [–1, 1] alternately. The behavior of the activation function will describe the characteristics of an artificial neuron model.
- The basic learning algorithm for a single layer perceptron repeats the following steps until the weights converge :

1. Select an input vector x from the training data set.
2. If the perceptron gives an incorrect response, modify all connection weights w_i according to

$$\Delta w_i = \eta\ t_i\ X_i$$

Where t_i is a target output and η is a learning state.

Q.10 Explain with diagram representation of neural network.

- Fig. Q.10.1 shows the neural network representation.

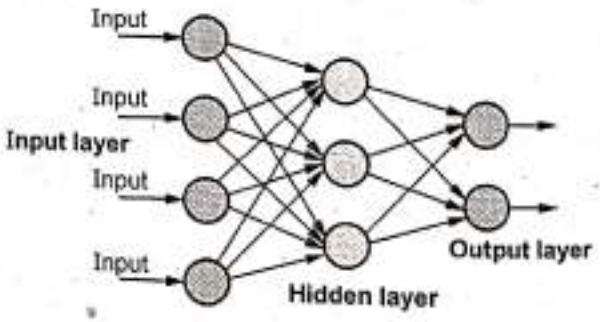


Fig. Q.10.1 Artificial neural network

- Neural Networks consists of many number of simple elements (neurons) connected between them in system. Whole system is able to solve of complex tasks and to learn for it like a natural brain.
- For user NN is black box with Input vector (source data) and Output vector (result).
- A Neural Network is usually structured into an input layer of neurons, one or more hidden layers and one output layer.

- Neurons belonging to adjecent layers are usually fully connected and the various types and architectures are identified both by the different topologies adopted for the connections as well by the choice of activation function.
- The values of the functions associated with the connections are called "weights".
- The whole game of using NNs is in the fact that, in order for the network to yield appropriate outputs for given inputs, the weight must be set to suitable values. The way this is obtained allows a further distinction among modes of operations.
- A neural network is a processing device, either an algorithm or actual hardware, whose design was motivated by the design and functioning of human brains and components thereof.
- Most neural networks have some sort of "training" rule whereby the weights of connections are adjusted on the basis of presented patterns.
- In other words, neural networks "learn" from examples, just like children learn to recognize dogs from examples of dogs, and exhibit some structural capability for generalization.
- Neural networks normally have great potential for parallelism, since the computations of the components are independent of each other.
- Neural networks are a different paradigm for computing :
  1. Von Neumann machines are based on the processing/memory abstraction of human information processing.
  2. Neural networks are based on the parallel architecture of animal brains.
- Neural networks are a form of multiprocessor computer system, with :

a. Simple processing elements
b. A high degree of interconnection
c. Simple scalar messages
d. Adaptive interaction between elements.

Q.11 What is appropriate problems for neural network ?

Ans. : • The backpropagation algorithm is the most commonly used ANN learning technique. It is appropriate for problems with the following characteristics :

1. Instances are represented by many attribute-value pairs.
2. The target function output may be discrete-valued, real-valued, or a vector of several real- or discrete-valued attributes.
3. The training examples may contain errors. Long training times are acceptable.
4. Fast evaluation of the learned target function may be required.
5. The ability for humans to understand the learned target function is not important.

2.2 : Perceptions

Q.12 What is perceptron ?

Ans. : • An arrangement of one input layer of McCulloch-Pitts neurons feeding forward to one output layer of McCulloch-Pitts neurons is known as a Perceptron.

- The perceptron is a feed - forward network with one output neuron that learns a separating hyper - plane in a pattern space.
- The "n" linear Fx neurons feed forward to one threshold output Fy neuron. The perceptron separates linearly set of patterns.

Q.13 Discuss the representable power of a perceptron. **☞[JNTU : Dec.-17, Marks 5]**

Ans. : • We can view the perceptron as representing a hyperplane decision surface in the n-dimensional space of instances.

- The perceptron outputs a1 for instances lying on one side of the hyperplane and outputs a-1 for instances lying on the other side.
- Some sets of positive and negative examples cannot be separated by any hyperplane. Those that can be separated are called linearly separable sets of examples.
- A single perceptron can be used to represent many Boolean functions. For example, if we assume Boolean values of 1 (true) and – 1 (false), then one way to use a two-input perceptron to implement the AND function is to set the weights.

- Consider two-input patterns (X_1, X_2) being classified into two classes as shown in Fig. Q.13.1. Each point with either symbol of x or 0 represents a pattern with a set of values (X_1, X_2).
- Each pattern is classified into one of two classes. Notice that these classes can be separated with a single line L. They are known as linearly separable patterns.
- Linear separability refers to the fact that classes of patterns with η-dimensional vector $x = (X_1, X_2, ..., X_n)$ can be separated with a single decision surface. In the case above, the line L represents the decision surface.
- If two classes of patterns can be separated by a decision boundary, represented by the linear equation then they are said to be linearly separable. The simple network can correctly classify any patterns.

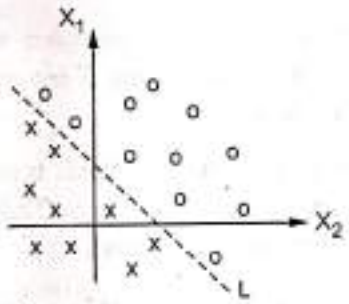


Fig. Q.13.1

- Decision boundary (i.e., W, b or q) of linearly separable classes can be determined either by some learning procedures or by solving linear equation systems based on representative patterns of each classes.
- If such a decision boundary does not exist, then the two classes are said to be linearly inseparable.
- Linearly inseparable problems cannot be solved by the simple network, more sophisticated architecture is needed.

Q.14 Discuss the decision surface of perceptron.

☞ [JNTU : Dec.-16, Marks 5]

Ans. : • A single perceptron can be used to represent many Boolean functions. For example, if we assume Boolean values of 1 (true) and – 1 (false), then on way to use a two-input perceptron to implement th AND function.

- Perceptron can represent all of the primitiv Boolean functions AND, OR, NAND (1 AND), an NOR (1 OR). Unfortunately, however, some Boolea functions cannot be represented by a singl perceptron, such as the XOR function whose valu is 1 if and only if $x_1 \neq x_2$.
- The decision surface represented by a two-inpu perceptron.

a) A set of training examples and the decisio surface of a perceptron that classifies the correctly.

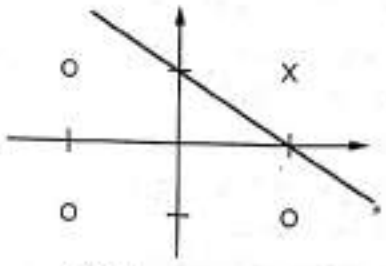
b) A set of training examples that is not linearl separable (i.e., that cannot be correctly classifie by any straight line).

- X_1 and X_2 are the Perceptron inputs. Positiv examples are indicated by "+", negative by "–".

1. Logical AND function

Patterns (bipolar)		
x_1	x_2	y
– 1	– 1	– 1
– 1	1	– 1
1	– 1	– 1
1	1	1

Decision boundary
$w_1 = 1$
$w_2 = 1$
$b = -1$
$q = 0$
$-1 + x_1 + x_2 = 0$



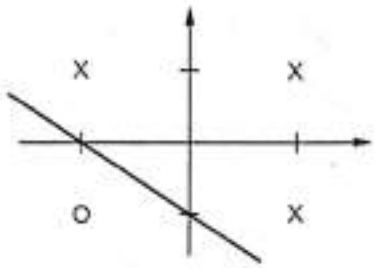
X : Class I (y = 1)
O : Class II (y = –1)

Fig. Q.14.1

2. Logical OR function

Patterns (bipolar)		
x_1	x_2	y
– 1	– 1	– 1
– 1	1	1
1	– 1	1
1	1	1

Decision boundary
$w_1 = 1$
$w_2 = 1$
$b = -1$
$q = 0$
$1+x_1 + x_2 = 0$



X : Class I (y = 1)
O : Class II (y = –1)

Fig. Q.14.2

Q.15 Explain gradient descent algorithm. What is steepest descent algorithm?

Ans. : • **Goal** : Solving minimization nonlinear problems through derivative information

- First and second derivatives of the objective function or the constraints play an important role in optimization. The first order derivatives are called the **gradient** and the second order derivatives are called the **Hessian matrix**.
- Derivative based optimization is also called **nonlinear**. Capable of determining search directions" according to an objective function's derivative information.
- Derivative based optimization methods are used for :
  1. Optimization of nonlinear neuro-fuzzy models
  2. Neural network learning
  3. Regression analysis in nonlinear models
- Basic descent methods are as follows :
  1. Steepest descent
  2. Newton-Raphson method

Gradient Descent :

- Gradient descent is a first-order optimization algorithm. To find a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point.
- Gradient descent is popular for very large-scale optimization problems because it is easy to implement, can handle black box functions, and each iteration is cheap.
- Given a differentiable scalar field f (x) and an initial guess x_1 , gradient descent iteratively moves the guess toward lower values of "f" by taking steps in the direction of the negative gradient – ∇ f (x).
- Locally, the negated gradient is the steepest descent direction, i.e., the direction that x would need to move in order to decrease "f" the fastest. The algorithm typically converges to a local minimum, but may rarely reach a saddle point, or not move at all if x_1 lies at a local maximum.
- The gradient will give the slope of the curve at that x and its direction will point to an increase in the function. So we change x in the opposite direction to lower the function value :

$$X_{k+1} = x_k - \lambda \nabla f(x_k)$$

The $\lambda > 0$ is a small number that forces the algorithm to make small jumps

Limitations of Gradient Descent :

- Gradient descent is relatively slow close to the minimum: technically, its asymptotic rate of convergence is inferior to many other methods.
- For poorly conditioned convex problems, gradient descent increasingly 'zigzags' as the gradients point nearly orthogonally to the shortest direction to a minimum point

Steepest Descent :

- Steepest descent is also known as gradient method.
- This method is based on first order Taylor series approximation of objective function. This method is also called saddle point method. Fig. Q.15.1 shows steepest descent method.
- The Steepest Descent is the simplest of the gradient methods. The choice of direction is where f decreases most quickly, which is in the direction

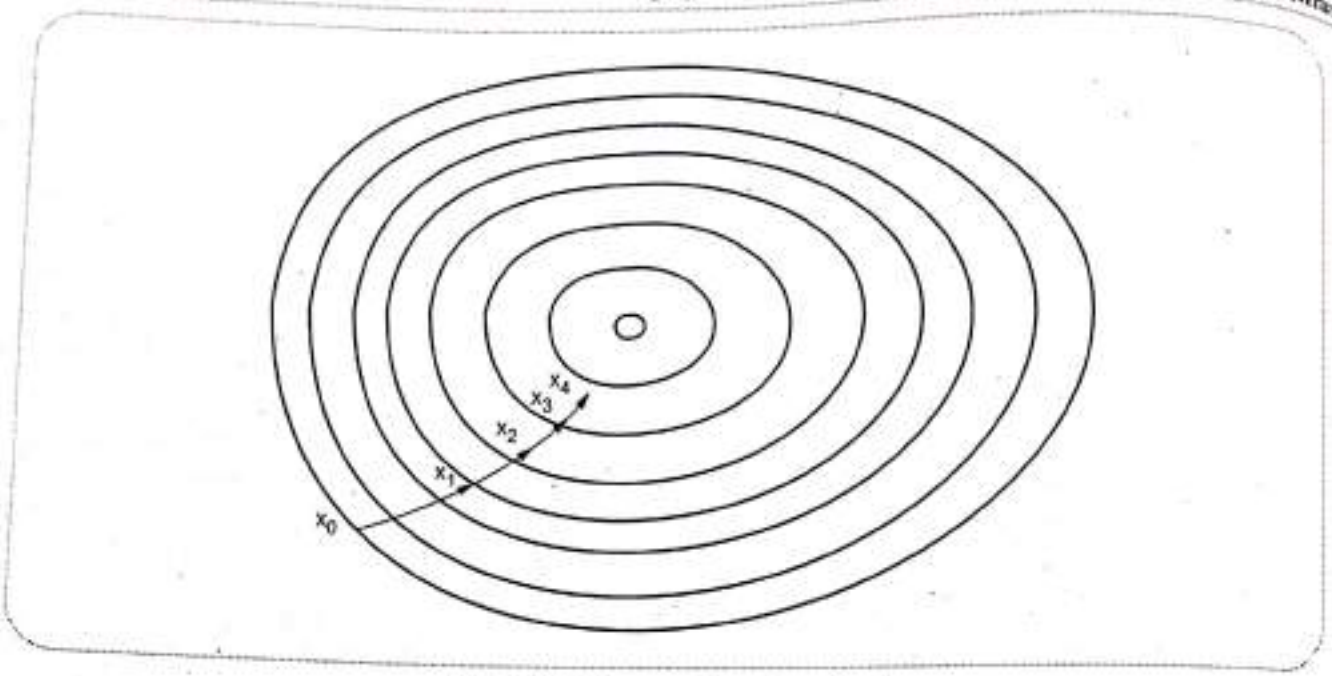


Fig. Q.15.1 Steepest descent method

opposite to $\nabla f(x_i)$. The search starts at an arbitrary point x0 and then go down the gradient, until reach close to the solution.

- The method of steepest descent is the discrete analogue of gradient descent, but the best move is computed using a local minimization rather than computing a gradient. It is typically able to converge in few steps but it is unable to escape local minima or plateaus in the objective function.
- The gradient is everywhere perpendicular to the contour lines. After each line minimization the new gradient is always orthogonal to the previous step direction. Consequently, the iterates tend to zig-zag down the valley in a very inefficient manner.
- The method of Steepest Descent is simple, easy to apply, and each iteration is fast. It also very stable; if the minimum points exist, the method is guaranteed to locate them after at least an infinite number of iterations.

Q.16 Explain delta learning rule for multiperceptron layer.

Ans. :

- An important generalization of the perceptron training algorithm was presented by Widrow and Hoff as the least mean square learning procedure also known as the delta rule.
- The learning rule was applied to the "**adaptive linear element**" also named Adaline.
- The perceptron learning rule uses the output of the threshold function for learning. The delta rule uses the net output without further mapping into output values –1 or +1.
- Fig. Q.16.1 shows adaline. (See Fig. Q.16.1 on next page).
- If the input conductances are denoted by w_i, where i = 0, 1, 2,.....n, and input and output signals by x_i and y, respectively, then the output of the central block is defined to be :

$$y = \sum_{i=1}^{n} w_i \; x_i + \theta$$

where $\theta = w_0$

- In a simple physical implementation this device consists of a set of controllable resistors connected to a circuit which can sum up currents caused by the input voltage signals. Usually the central block the summer is also followed by a quantizer which outputs either + 1 of – 1, depending on the polarity of the sum.
- The problem is to determine the coefficients w_i where i = 0, 1.....n, in such a way that the input output response is correct for a large number of arbitrarily chosen signal sets.

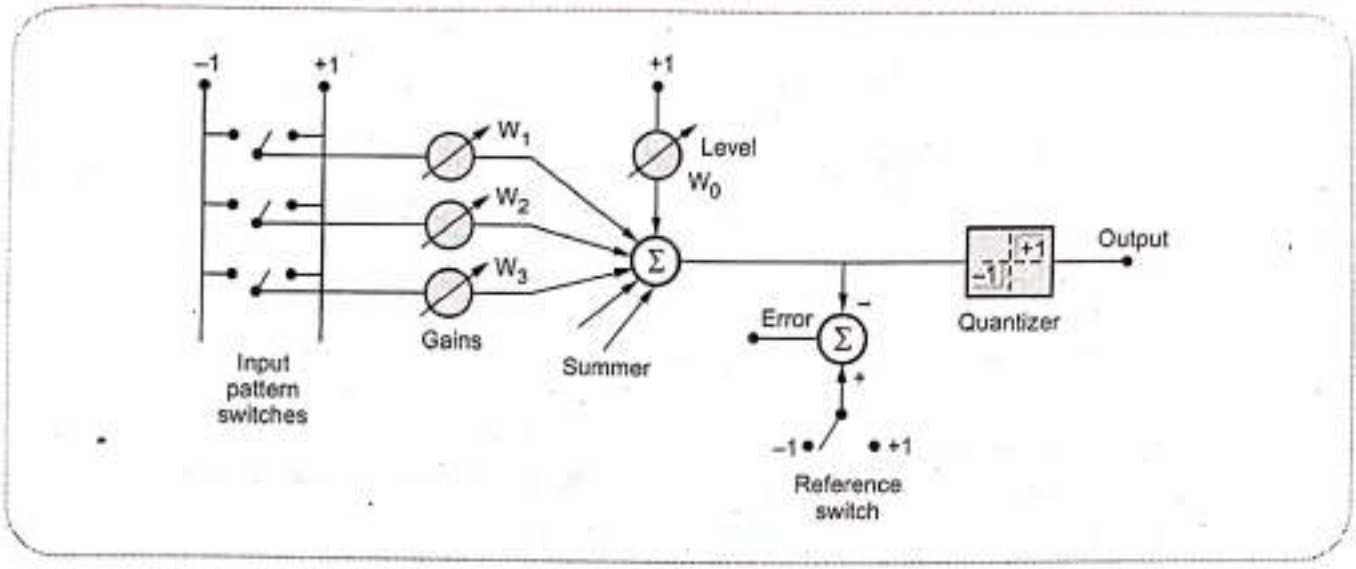


Fig. Q.16.1 Adaline

- If an exact mapping is not possible the average error must be minimized, for instance, in the sense of least squares.
- An adaptive operation means that there exists a mechanism by which the w_i can be adjusted, usually iteratively to attain the correct values.
- For the Adaline, Widrow introduced the delta rule to adjust the weights.
- For the p^{th} input-output pattern, the error measure of a single-output Adaline can be expressed as,

$$E_p = (t_p - o_p)^2$$

Where t_p = Target output

o_p = Actual output of the Adaline

- The derivation of E_p with respect to each weight w_i is

$$\frac{\partial E_p}{\partial w_i} = -2 (t_p - o_p) x_i$$

- To decrease E_p by gradient descent, the update formula for w_i on the p^{th} input-output pattern is

$$\Delta_p w_i = \eta (t_p - o_p) x_i$$

- The delta rule tries to minimize squared errors, it is also referred to as the **least mean square learning procedure** or **Widrow-Hoff learning rule.**
- **Features of the delta rule are as follows :**
  1. Simplicity
  2. Distributed learning : learning is not reliant on central control of the network.
  3. Online learning : weights are updated after presentation of each pattern.

Rules for Feedforward Multilayer Perceptron

- The training algorithm is called Error Back Propagation (EBP) training algorithm. If a submitted pattern provides an output far from desired value, the weights and thresholds are adjusted so that the current mean square classification error is reduced.
- The training is repeated for all patterns until the training set provide an acceptable overall error. Usually the mapping error is computed over the full training set.
- Error back propagation algorithm is working in two stages :
  1. The trained network operates feed-forward to obtain output of the network.
  2. The weight adjustment propagate backward from output layer through hidden layer toward input layer.

Q.17 What are differences between gradient descent and stochastic gradient descent ?

Ans. :

1. In standard gradient descent, the error is summed over all examples before updating weights, whereas in stochastic gradient descent weights are updated upon examining each training example.

2. Summing over multiple examples in standard gradient descent requires more computation per weight update step. On the other hand, because it uses the true gradient, standard gradient descent is often used with a larger step size per weight update than stochastic gradient descent.

2.3 : Multilayer Network and the Back-propagation Algorithm

Q.18 What is back propagation neural network ?

Ans. : Backpropagation is a training method used for a multi layer neural network. It is also called the generalized delta rule. It is a gradient descent method which minimizes the total squared error of the output computed by the net.

Q.19 List the training stages of a neural network by back propagation.

Ans. :

- The training of a neural network by back propagation takes place in three stages :

1. Feedforward of the input pattern
2. Calculation and Back propagation of the associated error.
3. Adjustments of the weights.

Q.20 Explain in brief architecture of multilayer feed-forward neural network.

Ans. : • A multilayer feed-forward neural network is a network consisting of multiple layers of units, all of which are adaptive.

- The network is not allowed to have cycles from later layers back to earlier layers, hence the name "feed-forward".
- Let us consider a network with a single complete hidden layer. i.e., the network consists of some input nodes, some output nodes, and a set of hidden nodes.
- Every hidden node takes inputs from each of the input nodes, and feeds into each of the output nodes.
- Fig. Q.20.1 shows multilayer feed forward neural network.

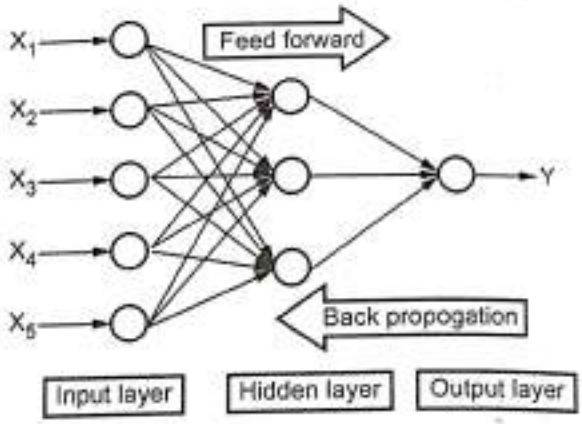


Fig. Q.20.1 Multilayer feed forward neural network

- This structure is called multilayer because it has a layer of processing units (i.e., the hidden units) in addition to the output units.
- These networks are called feedforward because the output from one layer of neurons feeds forward into the next layer of neurons. There are never any backward connections, and connections never skip a layer.
- Each connection between nodes has a weight associated with it. In addition, there is a special weight (called w_0) that feeds into every node at the hidden layer and a special weight (called z_0) that feeds into every node at the output layer.
- These weights are called the bias, and set the thresholding values for the nodes. Initially, all of the weights are set to some small random values near zero.
- Every node in the hidden layer and in the output layer processes its weighted input to produce an output. This can be done slightly differently at the hidden layer, compared to the output layer.
- **Input units :** The input data you provide your network comes through the input units. No processing takes place in an input unit, it simply feeds data into the system.
- **Hidden units :** The connections coming out of an input unit have weights associated with them. A weight going to hidden unit z_h from input unit x_i would be labeled w_{hj}. The bias input node (x_0) is connected to all the hidden units, with weights w_{hj}.
- Each hidden node calculates the weighted sum of its inputs and applies a thresholding function to

determine the output of the hidden node. The weighted sum of the inputs for hidden node z_h is calculated as :

$$\sum_{j=0}^{d} w_{hj} x_j$$

- The thresholding function applied at the hidden node is typically either a step function or a sigmoid function. The sigmoid function is sometimes called the "squashing" function, because it squashes its input (i.e., a) to a value between 0 and 1.
- In multi-layer feed forward neural networks, the sigmoid activation function, defined by $g(x) = \frac{1}{1+e^{-x}}$ is normally used.
- **The output layer :** Functionally just like the hidden layers. Outputs are passed on to the world outside the neural network.

Q.21 How does the network learn ?

Ans. : • The training samples are passed through the network and the output obtained from the network is compared with the actual output.

- This error is used to change the weights of the neurons such that the error decreases gradually.
- This is done using the Backpropagation algorithm, also called backprop. Iteratively passing batches of data through the network and updating the weights, so that the error is decreased, is known as Stochastic Gradient Descent (SGD).
- The amount by which the weights are changed is determined by a parameter called learning rate.

Q.22 Why multiLayer perceptron neural network ?

Ans. : • Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques.

- A trained neural network can be thought of as an "expert" in the category of information it has been given to analyse.
- This expert can then be used to provide projections given new situations of interest and answer "what if" questions.
- Other advantages include :
  1. Adaptive learning : An ability to learn how to do tasks based on the data given for training or initial experience.
  2. One of the preferred techniques for gesture recognition.
  3. MLP/Neural networks do not make any assumption regarding the underlying probability density functions or other probabilistic information about the pattern classes under consideration in comparison to other probability based models.
  4. They yield the required decision function directly via training.
  5. A two layer backpropagation network with sufficient hidden nodes has been proven to be a universal approximator

Q.23 Explain backpropagation learning rule.

Ans. : • The **net input** of a node is defined as the weighted sum of the incoming signals plus a bias term. Fig. Q.23.1 shows the backpropagation MLP for node j. The net input and output of node j is as follows :

$$\bar{X}_j = \sum_i + W_{ij} + W_j$$

$$x_j = f(\bar{X}_j) = \frac{1}{1+\exp(-\bar{X}_j)}$$

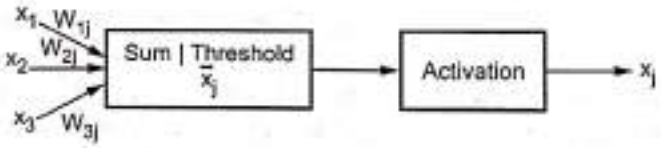


Fig. Q.23.1 Backpropagation MLP for node j

Where x_1 is the output of node i located in any one of the previous layers,

W_{ij} is the weight associated with the link corresponding nodes i and j.

W_j is the bias of node j.

- Internal parameters associated with each node j is the weight W_{ij}. So changing the weights of the node will change the behaviour of the whole back propagation MLP.

- Fig. Q.23.2 shows two layer back propagation MLP.

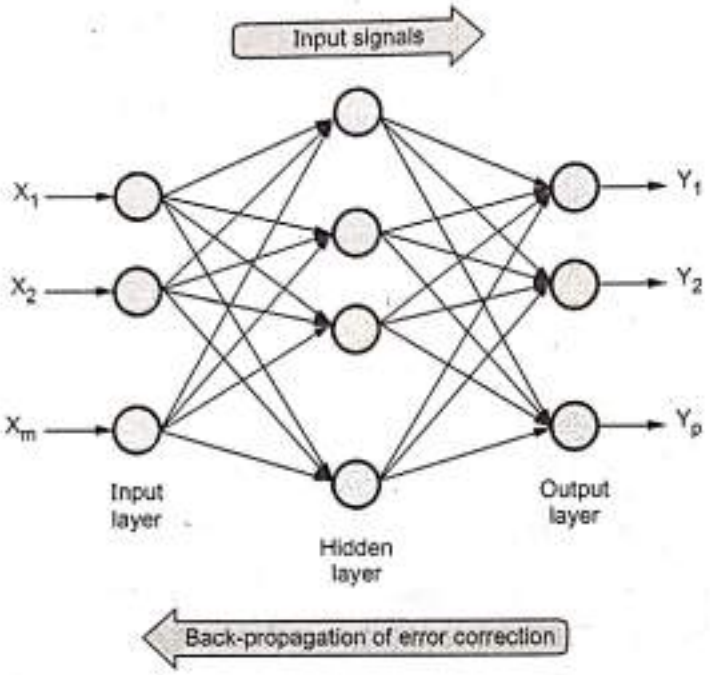


Fig. Q.23.2

- The above back propagation MLP will refer to as a 3-4-3 network, corresponding to the number of nodes in each layer.
- The backward error propagation also known as the Backpropagation (BP) or the Generalized Delta Rule (GDR). A squared error measure for the p^{th} input-output pair is defined as

$$E_p = \sum_k (d_k - x_k)^2$$

Where d_k is the desired output for node k and x_k is the actual output for node k when the input part of the p^{th} data pair is presented.

Q.24 Write the characteristics and applications of error back propagation algorithm.

Ans. : Characteristics :

- It is an algorithm for supervised learning of artificial neural networks using gradient descent.
- Learns weights for a multilayer network, given a fixed set of units and interconnections.
- In multilayer networks the error surface can have multiple minima, but in practice backpropagation has produced excellent results in many real-world applications.
- The algorithm is for two layers of sigmoid units and does stochastic gradient descent.
- It uses gradient descent to minimize the squashed error between the network outputs and the target values for these outputs.

Applications :

- The fast development of artificial satellite technology has increased the importance of three dimensional (3D) positioning and therefore, satellite geodesy.
- Particularly, the Global Positioning System (GPS) provides more practical, rapid, precise and continuous positioning results anywhere on the Earth in geodetic applications when compared to the traditional terrestrial positioning methods.
- Due to the increasing use of GPS positioning techniques, a great attention has been paid to the precise determination of local/regional geoids, aiming at replacing the geometric leveling with GPS measurements.
- Therefore, BPANN method is easily programmable with decreased and increased number of reference points when generating a local GPS, it performs a flexible modelling.
- Also, BPANN method is open to updating which could be accepted as an important advantage. Thus, it is believed that BPANN method is more convenient for generating local GPS when compared to other methods.

Q.25 List out merits and demerits of EBP.

Ans. : Merits/Strength :

1. Computing time is reduced if weight chosen are small at the beginning.
2. It minimize the error
3. Batch update of weight exist, which provide smoothing effects on the weight correction.
4. Simple method and easy for implementation.
5. Minimum of the error function in weight space
6. Standard method and generally work well

Demerits :

1. Training may sometime cause temporal instability to the system.
2. For complex problem, it takes lot of times.

3. Selection of number of hidden node in the network is problem.
4. Backpropagation learning does not require normalization of input vectors; however, normalization could improve performance
5. It can get stuck in local minima resulting in sub-optimal solutions.
6. Slow and inefficient.

Q.26 Discuss performance issue of EBP.

Ans. : • Computational Efficiency is main aspect of back-propagation. Number of operations to compute derivatives of error function scales with total number W of weights and biases.

- Single evaluation of error function for a single input requires O(W) operations for large W. Compute derivatives using method of finite differences.

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{E_n(w_{ji}+\varepsilon) - E_n(w_{ji})}{\varepsilon} + O(\varepsilon)$$

where $\varepsilon << 1$

- Accuracy can be improved by making ε smaller until round-off problems arise.
- Accuracy can be improved by using central differences.

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{E_n(w_{ji}+\varepsilon) - E_n(w_{ji}-\varepsilon)}{2\varepsilon} + O(\varepsilon^2)$$

- This is $O(W^2)$.
- Generalization means performance on input patterns, i.e. input patterns which were not among the patterns on which the network was trained.
- If you train for too long, you can often get the sum-squared error very low, by over-fitting the training data you get a network which performs very well on the training data, but not as well as it could on unseen data.
- By stopping training earlier, one hopes that the network will have learned the broad rules of the problem, but not bent itself into the shape of some of the more idiosyncratic (perhaps even noisy) training patterns.

2.4 : Remarks on the Back-propagation Algorithm

Q.27 Why does overfitting tend to occur during later iterations, but not during earlier iterations ?

Ans. : • Consider that network weights are initialized to small random values. With weights of nearly identical value, only very smooth decision surfaces are describable.

- As training proceeds, some weights begin to grow in order to reduce the error over the training data, and the complexity of the learned decision surface increases.
- Thus, the effective complexity of the hypotheses that can be reached by **BACKPROPAGATION** increases with the number of weight-tuning iterations.
- Given enough weight-tuning iterations, **BACKPROPAGATION** often be able to create overly complex decision surfaces that fit noise in the training data or unrepresentative characteristics of the particular training sample.
- This overfitting problem is analogous to the overfitting problem in decision tree learning.

Q.28 Explain expressive capabilities of artificial neural networks. ☞ **[JNTU : Dec.-16, Marks 5]**

Ans. : Boolean functions :

- Every Boolean function can be represented by network with a single hidden layer
- But that might require an exponential (in the number of inputs) hidden units

Continuous functions :

- Every bounded continuous function can be approximated with arbitrarily small error by a network with one hidden layer.
- Any function can be approximated to arbitrary accuracy by a network with two hidden layers.

2.5 : An Illustrate Example : Face Recognition

Q.29 Explain an illustrate example of Face Recognition.

Ans. : • Face Recognition is a multi-class classification problem in which face is classified as belonging to any subject. Face recognition is substantially different from classical pattern recognition problems, such as object recognition.

- The shapes of the objects are usually different in an object recognition task, while in face recognition one always identifies objects with the same basic shape.
- The learning task involves classifying camera images of faces of various people in various poses.
- A variety of target functions can be learned from this image data. For example, given an image as input we could train an ANN to output the identity of the person, the direction in which the person is facing, the gender of the person, whether or not they are wearing sunglasses, etc.
- All of these target functions can be learned to high accuracy from this image data, and the reader is encouraged to try out these experiments.
- **Input encoding :** Given that the ANN input is to be some representation of the image, one key design choice is how to encode this image. For example, we could pre-process the image to extract edges, regions of uniform intensity, or other local image features, then input these features to the network.
- **Output encoding :** The ANN must output one of four values indicating the direction in which the person is looking (left, right, up, or straight).
- Use four distinct output units for representing one of the four possible face directions, with the highest-valued output taken as the network prediction. This is often called a 1-of-n output encoding.
- **Network graph structure :** Backpropagation can be applied to any acyclic directed graph of sigmoid units. The most common network structure is a layered network with feedforward connections from every unit in one layer to every unit in the next.
- Network weights in the output units were initialized to small random values. However, input unit weights were initialized to zero, because this yields much more intelligible visualizations of the learned weights, without any noticeable impact on generalization accuracy.

2.6 : Advanced Topics in Artificial Neural Network

Q.30 What is RNN ?

Ans. : • A recurrent neural network (RNN) is a type of artificial neural network commonly used in speech recognition and Natural Language Processing (NLP).

- RNNs are designed to recognize a data's sequential characteristics and use patterns to predict the next likely scenario. RNNs are used in deep learning and in the development of models that simulate the activity of neurons in the human brain.
- The simplest form of fully recurrent neural network is an MLP with the previous set of hidden unit activations feeding back into the network along with the inputs.

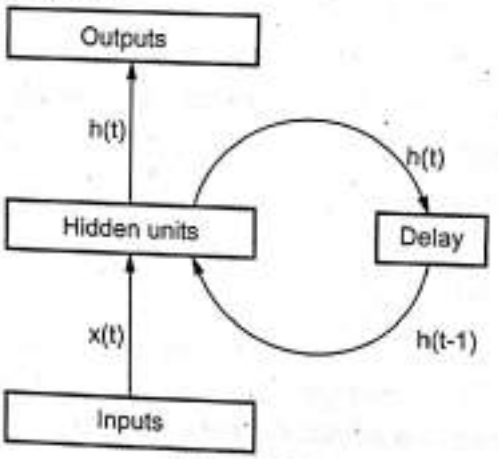


Fig. Q.30.1

- Note that the time t has to be discretized, with the activations updated at each time step.
- The time scale might correspond to the operation of real neurons, or for artificial systems any time step size appropriate for the given problem can be used. A delay unit needs to be introduced to hold activations until they are processed at the next time step.
- If the neural network inputs and outputs are the vectors x(t) and y(t), the three connection weight matrices are W_{IH}, W_{HH} and W_{HO}, and the hidden

and output unit activation functions are f_H and f_O, the behaviour of the recurrent network can be described as a dynamical system by the pair of non-linear matrix equations :

$$h(t) = f_H(W_{IH}x(t) + W_{HH}h(t-1))$$

$$y(t) = f_O(W_{HO}h(t))$$

- In general, the state of a dynamical system is a set of values that summarizes all the information about the past behaviour of the system that is necessary to provide a unique description of its future behaviour, apart from the effect of any external factors.
- In this case the state is defined by the set of hidden unit activations h(t).
- Thus, in addition to the input and output spaces, there is also a state space. The order of the dynamical system is the dimensionality of the state space, the number of hidden units.

2.7 : Evaluation Hypotheses

Q.31 What is hypotheses ?

Ans. : • A hypothesis is a statement of a relationship between two or more variables.

- The solution(s) to machine learning tasks are often called **hypotheses**, because they can be expressed as a hypothesis that the observed positives and negatives for a categorization is explained by the concept learned for the solution.
- The hypotheses have to be represented in some representation scheme, and, as usual with AI tasks, this will have a big effect on many aspects of the learning methods.

Q.32 Define sample error.

[JNTU : Dec.-16, Marks 2]

Ans. : The sample error ($error_S(h)$) of hypothesis h with respect to target function f and data sample S is

$$error_s(h) = \frac{1}{n}\sum_{x \in S}\delta(f(x), h(x))$$

Where n is the number of examples in S, and the quantity $\delta(f(x), h(x))$ is 1 if $f(x) \neq h(x)$, and 0 otherwise.

Q.33 Differentiate between sample error and true error.

[JNTU : Dec.-17, Marks 3]

Ans. :

- The *sample error* of a hypothesis with respect to some sample S of instances drawn from X is the fraction of S that it misclassifies.
- The **true error** of a hypothesis is the probability that it will misclassify a single randomly drawn instance from the distribution D.
- **Sample error :**

$$error_s(h) = \frac{1}{n}\sum_{x \in S}\delta(f(x), h(x))$$

- **True error :**

$$error_D(h) = \Pr_{x \in D}[f(x) \neq h(x)]$$

Q.34 What are the reasons for using hypotheses ?

Ans. : • Learning from a limited-size database indicating the effectiveness of different medical treatments, it is important to understand as precisely as possible the accuracy of the learned hypotheses.

- The evaluating hypotheses are an integral component of many learning methods.
- It is important to understand the likely errors inherent in estimating the accuracy of the pruned and unpruned tree.
- Estimating the accuracy of a hypothesis is relatively straightforward when data is plentiful.
- An estimator is any random variable used to estimate some parameter of the underlying population from which a sample is drawn.

1. The estimation bias of an estimator Y for an arbitrary parameter p is $E[Y] - p$, If the estimation bias is 0, then Y is an unbiased estimator for p.
2. The variance of an estimator Y for an arbitrary parameter p is simply the variance of Y.

Q.35 Explain confidence intervals for discrete-valued hypotheses.

Ans. : • To estimate the true error for some discrete valued hypothesis (h), based on its observed sample error over a sample (S), where the sample S contains n examples drawn independent of one another, and independent of h, according to the probability

distribution V hypothesis h commits r errors over these n examples.

- Statistical theory allows to make the following assertions :
  1. Given no other information, the most probable value of $error_D(h)$ is $error_s(h)$ and sample equal to 30.
  2. With approximately 95 % probability, the true error $error_D(h)$ lies in the interval

$$error_s(h) \pm 1.96\sqrt{\frac{error_s(h)(1-error_s(h))}{n}}$$

Hypothesis h misclassifies 12 of the 40 examples in S

$$error_s(h) = \frac{12}{40} = .30$$

With approximately 95% probability, $error_D(h)$ lies in interval

$$error_s(h) \pm 1.96\sqrt{\frac{error_s(h)(1-error_s(h))}{n}}$$

$$= .30 \pm 1.96\sqrt{\frac{.30\times.70}{40}}$$

$$= .30 \pm 1.96 \times .072$$

$$= .30 \pm .14$$

- The above expression for the 95% confidence interval can be generalized to any desired confidence level. The constant 1.96 is used in case we desire a 95 % confidence interval.
- A different constant (Z_N) is used to calculate the N % confidence interval. The general expression for approximate N % confidence intervals for $error_D(h)$ is

$$bias \equiv E[error_s(h)] - error_D(h)$$

$$error_s(h) \pm Z_N\sqrt{\frac{error_s(h)(1-error_s(h))}{n}}$$

where the constant Z_N is chosen depending on the desired confidence level, using the values of Z_N given.

- **Bias :** If *S* is training set, $error_s(h)$ is optimistically biased.
- **Variance :** Even with unbiased *S*, $error_s(h)$ may still vary from $error_D(h)$.

2.8 : Basic of Sampling Theory

Q.36 What is random variable ?

Ans. : A function whose domain is a sample spac and whose range is a some set of real numbers i called random variables. Let a random variable X(A represent the functional relationship between random event A and a real number. A randor variable is a mapping from the sample space to th set of real numbers.

Q.37 What is Probability Distribution ?

Ans. : The behavior of a random variable i characterized by its probability distribution, that i by the way probabilities are distributed over th values it assumes. A probability mass function an two ways to characterize this distribution for discrete random variable.

Q.38 Define central limit theorem.

Ans. : The central limit theorem is a theorem statin that the sum of a large number of independen identically distributed random variable approximately follows a normal distribution.

Q.39 What is binomial distribution ? Explain its properties. Where binomial distribution is apply ?

Ans. :

- The Binomial distribution gives the general form o the probability distribution for the random variabl r, whether it represents the number of heads in coin tosses or the number of hypothesis errors in sample of n examples.
- The detailed form of the Binomial distributio depends on the specific sample size n and th specific probability p or $error_D(h)$.
- **Binomial distribution applies as follows :**
  1. There is a base, or underlying, experiment whos outcome can be described by a random variabl (Y). The random variable can take on tw possible values.
  2. The probability that Y = 1 on any single trial o the underlying experiment is given by som constant p, independent of the outcome of an other experiment.

3. A series of n independent trials of the underlying experiment is performed, producing the sequence of independent, identically distributed random variables $Y_1, Y_2, \ldots, Y_n$.

- Binomial means 'two numbers'.
- The outcomes of health research are often measured by whether they have occurred or not. For example, recovered from disease, admitted to hospital, died etc.
- The binomial distribution occurs in games of chance, quality inspection, opinion polls, medicine and so on.
- It may be modelled by assuming that the number of events 'n' has a binomial distribution with a fixed probability of event p. Binomial distribution is distribution for a series of Bernoulli trials.

- **Properties of binomial distribution :**

1. Experiment consist of n identical trials.
2. Each trial has only two outcomes.
3. The probability of one outcome is p and the other is $q = 1 - p$.
4. The trials are independent.
5. We are interested in x, the number of success observed during the n trials.

- Binomial distribution written as B (n, p) where n is the total number of events and p = probability of an event.
- A Binomial distribution gives the probability of observing r heads in a sample of n independent coin tosses, when the probability of heads on a single coin toss is p. It is defined by the probability function

$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

The mean μ (mu) of the binomial distribution is

$$\mu = np$$

The variance is,

$$\sigma^2 = npq$$

The mean and variance of binomial distribution with parameters (n, p) are given as,

$$\text{Mean} = \mu = E(X)$$

$$= \sum_{i=1}^{n} E(X_i)$$

$$= np$$

Q.40 Explain the relation between estimators, bias, and variance.

Ans. : We have :

$$\text{error}_s(h) = \frac{r}{n}$$

$$\text{error}_D(h) = p$$

Where n = Number of instances in the sample S,

r = Number of instances from S misclassified by h,

p = The probability of misclassifying a single instance drawn from D.

- Point estimation is the attempt to provide the single best prediction of some quantity of interest.
- **Is errors(h) an unbiased estimator for $\text{error}_D(h)$?**

  Yes, because for a Binomial distribution the expected value of r is equal to np. It follows, given that n is a constant, that the expected value of r /n is p.
- Bias and Variance measure two different sources of error of an estimator. Bias measures the expected deviation from the true value of the function or parameter.
- Variance provides a measure of the expected deviation that any particular sampling of the data is likely to cause.

2.9 : General Approach for Deriving Confidence Intervals

Q.41 Discuss briefly about confidence intervals.

Ans. : • An interval estimate is an interval or range of values, used to estimate a population parameter.

- A confidence interval estimate of a parameter consists of an interval of numbers along with a probability that the interval contains the unknown parameter.

- An interval estimate consists of two numerical values that, with a specified degree of confidence, we feel includes the parameter being estimated.
- An estimator is a rule or formula that tells how to compute the estimate. Estimators are unbiased if they predict well the value in the population.
- The sampled population is the population from which we actually draw the sample. The target population is the population about which we wish to make an inference.
- The sampled population and target populations may or may not be the same. When they are the same, it is possible to use statistical inference procedures to make conclusions about the target population.
- If the sample and target populations are different, conclusions can be made about the target population only on the basis of non-statistical considerations. The strict validity of statistical procedures depends on the assumption of random samples.
- **Assume N(0,1)**
- **We are interested in :**
  1. Finding the symmetric interval **around the mean** such that the probability of seeing a sample from it is p
  2. Measuring the distance of end points from 0 in terms of $\sigma = 1$.
- One common way to describe the uncertainty associated with an estimate is to give an interval within which the true value is expected to fall, along with the probability with which it is expected to fall into this interval. Such estimates are called confidence **interval** estimates.
- Fig. Q.41.1 shows confidence intervals.
- Probability mass under the normal curve for a symmetric interval around the mean is invariant when interval distances are measured in terms of the standard deviation.
- Problem : But typically the variance is not known
- Solution : Estimate the variance from the sample

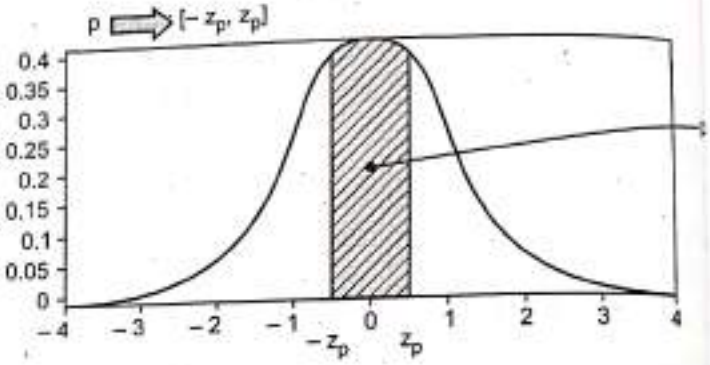


Fig. Q.41.1 Confidence intervals

$$S_n = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}}$$

- Assume the sample mean falls into the interval centered at the mean :

$$\overline{X} \in \left[\mu - t_p \frac{S_n}{\sqrt{n}}, \mu + t_p \frac{S_n}{\sqrt{n}}\right]$$

- Or equivalently that the mean falls into the interval centered around the sample mean :

$$\mu \in \left[\overline{X} - t_p \frac{S_n}{\sqrt{n}}, \overline{X} + t_p \frac{S_n}{\sqrt{n}}\right]$$

- **This happens with some probability p that depends on t_p**
- **Let,**

$$t = \frac{\overline{X} - \mu}{S_n}\sqrt{n}$$

- The difference from the known variance case :
  - t is not normally distributed, instead it follows a **Student distribution** (t distribution).
  - Student distribution has one additional parameter : the **degree of freedom.**
  - **For**

$$S_n = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}}$$

t has n – 1 degrees of freedom

$$t(n-1) = \frac{\overline{X} - \mu}{S_n}\sqrt{n} = t \text{ distribution } (n-1)$$

Q.42 Discuss general approach for deriving confidence intervals.

Ans. : The general process includes the following steps :

1. Identify the underlying population parameter p to be estimated, for example, $error_D(h)$.
2. Define the estimator Y. It is desirable to choose a minimum variance, unbiased estimator.
3. Determine the probability distribution D_Y that governs the estimator Y, including its mean and variance.
4. Determine the N% confidence interval by finding thresholds L and U such that N% of the mass in the probability distribution D_Y falls between L and U.

Q.43 Explain central limit theorem.

Ans. :

- The sampling distribution of the sample mean, x is approximated by a normal distribution when the sample is a simple random sample and the sample size, n, is large.
- In this case, the mean of the sampling distribution is the population mean (μ) and the standard deviation of the sampling distribution is the population standard deviation (σ), divided by the square root of the sample size. The later is referred to as the standard error of the mean.
- A sample size of 100 or more elements is generally considered sufficient to permit using the CLT. If the population from which the sample is drawn is symmetrically distributed, n > 30 may be sufficient to use the CLT.
- The central limit theorem states that the mean of the sampling distribution of the mean will be the unknown population mean. The standard deviation of the sampling distribution of the mean is called the standard error.
- In fact, it is just another standard deviation, we just call it the standard error so we know we're talking about the standard deviation of the sample means instead of the standard deviation of the raw data. The standard deviation of data is the average distance values are from the mean.

2.10 : Difference in Error of Two Hypotheses

Q.44 What is hypothesis testing ? What is the basic assumption of hypothesis testing ?

Ans. :

- A statistical hypothesis test is a procedure for deciding between two possible statements about a population. The phrase significance test means the same thing as the phrase "hypothesis test."
- A hypothesis test is a statistical method that uses sample data to evaluate a hypothesis about a population.
- The general goal of a hypothesis test is to rule out chance as a plausible explanation for the results from a research study.
- The goal in hypothesis testing is to analyze a sample in an attempt to distinguish between population characteristics that are likely to occur and population characteristics that are **unlikely** to occur.

Basic assumption of hypothesis testing

1. If the treatment has any effect, it is simply to add or subtract a constant amount to each individual's score.
2. Remember that adding or subtracting constant changes the mean, but not the shape of the distribution for the population and/or the standard deviation.
3. The population after treatment has the same shape and standard deviation as the population prior to treatment.
4. If the individuals in the sample are noticeably different from the individuals in the original population, we have evidence that the treatment has an effect.

Q.45 Explain steps in hypothesis testing.

Ans. : Step 1 : Formulate the hypothesis

- A null hypothesis is a statement of the status quo, one of no difference or no effect. If the null hypothesis is not rejected, no change will be made.
- An alternative hypothesis is one in which some difference or effect is expected.

- The null hypothesis refers to a specified value of the population parameter, not a sample statistic.

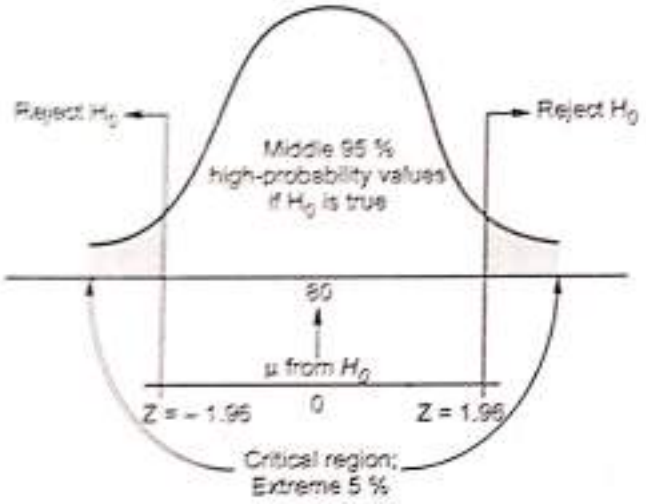


Fig. Q.45.1

Step 2 : Select an appropriate test

- The test statistic measures how close the sample has come to the null hypothesis.
- The test statistic often follows a well-known distribution (e.g., normal, t, or chi-square)
- Calculate Z statistic.

Step 3 : Choose level of significance

Type 1 error

- Occurs if the null hypothesis is rejected when it is in fact true.
- The probability of type I error (α) is also called the **level of significance.**

Type II error

- Occurs if the null hypothesis is not rejected when it is in fact false.
- The probability of type II error is denoted by β.
- Unlike α, which is specified by the researcher, the magnitude of β depends on the actual value of the population parameter (proportion).
- It is necessary to balance the two types of errors.
- The power of test is the probability $(1-\beta)$ of rejecting the null hypothesis when it is fake and should be rejected. Although β is unknown, it is related to α.

Step 4 : Collect data and calculate test statistic

- The required data are collected and the value of the test statistic z can be calculated as follows :

$$Z_{cal} = \frac{\hat{P} - \pi}{\sigma_p}$$

Step 5 : Determine probability value/critical value

- Using standard normal tables.
- Note, in determining the critical value of the test statistic, the area to the right of the critical value is either α or $\alpha/2$. It is α for a one-tall test and $\alpha/2$ for a two-tail test.
- If the probability associated with the calculated value of the test statistic (Z_{cal}) is less than the level of significance (α), the null hypothesis is rejected.
- Alternatively, if the calculated value of the best statistic is greater than the critical value of the best statistic (z_α), the null hypothesis is rejected.

Q.46 What is P-Value ? How it helps in hypothesis testing ?

Ans. : P-value and hypothesis testing

- As an alternative approach to the rejection/acceptance-region approach, we can calculate a probability related to the test statistic, called P-value and base our decision of rejection/acceptance on the magnitude of the P-value.
- P-value is the probability to observe a value of the test statistic as extreme as the one observed, if the null hypothesis is not true and hence should be rejected.

In a hypothesis testing problem :

a) The null hypothesis will not be rejected unless the data are not unusual (given that the hypothesis is true).

b) The null hypothesis will not be rejected the P-value indicates the data are very unusual (given that the hypothesis is true).

c) The null hypothesis will not be rejected only if the probability of observing the data provides convincing confidence that it is true.

d) The null hypothesis is also called the research hypothesis ; the alternative hypothesis often represents the status quo.

e) The null hypothesis is the hypothesis that we would like to prove ; the alternative hypothesis is called the research hypothesis.

2.11 : Comparing Learning Algorithm

Q.47 Discuss paired-t test.

Ans. : • Collect data in pairs.

- Example : Given training set D_{Train} and a test set DTest, train **both** learning algorithms on D_{Train} and then test their accuracies on D_{Test}.
- Suppose n paired measurements have been made.
- Assume the measurements are independent and the measurements for each algorithm follow a normal distribution
- The test statistic T_0 will follow a t-distribution with n-1 degrees of freedom

Q.48 Explain type-I and type-II errors.

Ans. :

Type I error

- The choice of what significance level to use (.05, .01, or lower or higher) is the difficult choice. If you decide to accept the 0.05 level of confidence, which requires a smaller "t value", you can more easily reject the null hypothesis and declare that there is a statistically significant difference between the means than if you select the 0.01 level, but you will be wrong 5 percent of the time. **This is the type I error.**
- A **type I error** occurs when the sample data appear to show a treatment effect when, in fact, there is none.
- Type I errors are caused by unusual, unrepresentative samples, falling in the critical region even though the treatment has no effect. The hypothesis test is structured so that type I errors are very unlikely; specifically, the probability of a type I error is equal to the alpha level.

The α level

- The α level is also known as the level of significance and known as type I error. It determines the risk of a false positive finding.
- The probability that a result would be produced by chance (sampling error or random error) alone. The commonly used levels of significance (α) :

1. α = .05 (most used) : 5 % or 5 out of every 100 results would be due to chance
2. α = .01 : 1 % or 1 out of every 100 results would be due to chance
3. α = .001 : 0.1 % or 1 out of every 1000 results would be due to chance

Select α **level** for two-tailed tests : Two-tailed tests hypothesize the presence of a difference, but not a particular direction for the difference between a sample mean (M) and a population mean (μ).

$H_0 : M = \mu$ $H_1 : M \neq \mu$

α level	Z score
0.05	± 1.96
0.01	± 2.58
0.001	± 3.30

Type II error

- On the other hand, if you select the 0.01 value, you will be wrong only 1 percent of the time. But since the .01 value requires a larger " t value", you will less often be able to reject the null hypothesis and say that there is a statistically significant difference between the means when in fact that is the case. **This is the type II error.**
- A **type II** error occurs when the sample does not appear to have been affected by the treatment when, in fact, the treatment does have an effect. In this case, the researcher will fail to reject the null hypothesis and falsely conclude that the treatment does not have an effect.
- Type II errors are commonly the result of a very small treatment effect. Although the treatment does have an effect, it is not large enough to show up in the research study.
- Type II error is also known as beta error (β). It is defined by the probability of false negatives.

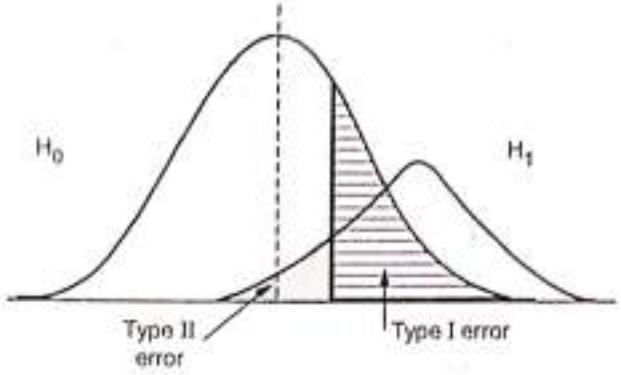


Fig. Q.48.1 Type I and II error

- An error made by accepting or retaining a false null hypothesis (H_0). It stated simply, you fail to reject a false null hypothesis (H_0) and claim that a relationship does not exist when it does exist.

Fill in the Blanks for Mid Term Exam

Q.1 The basic processing elements of neural networks are called _____.

Q.2 The _____ function used for a back propagation neural network can be either a bipolar sigmoid or a binary sigmoid.

Q.3 Backpropagation is a training method used for a _____ neural network.

Q.4 The _____ is a kind of a single layer artificial network with only one neuron.

Q.5 The process of weight adaptation is called _____.

Q.6 One successful method for finding high accuracy hypotheses is a technique called _____. [JNTU : Aug.-16, Feb.-17]

Q.7 _____ learning methods provide a robust approach to approximating real-valued, discrete-valued, and vector-valued target functions. [JNTU : Aug.-16, Feb.-17]

Q.8 Perceptrons can represent all of the primitive _____functions. [JNTU : Aug.-16, Feb.-17]

Q.9 Theoretical results have been developed that characterize the fundamental relationship among the number of _____ examples observed. [JNTU : Feb.-17]

Multiple Choice Questions for Mid Term Exam

Q.1 Training Perceptron is based on ----------

- a supervised learning technique
- b unsupervised learning
- c reinforced learning
- d stochastic learning

Q.2 What is perceptron in Neural network ?

- a It is an auto-associative neural network
- b It is a double layer auto-associative neural network
- c It is a single layer feed-forward neural network with pre-processing
- d It is a neural network that contains feedback

Q.3 Application of Neural Network includes ___

- a Pattern Recognition
- b Classification
- c Clustering
- d All of these

Q.4 Neural Networks are complex _____ with many parameters.

- a linear Functions
- b nonlinear Functions
- c discrete Functions
- d exponential Functions

Q.5 What is backpropagation ?

- a It is another name given to the curvy function in the perceptron
- b It is the transmission of error back through the network to adjust the inputs
- c It is the transmission of error back through the network to allow weights to be adjusted so that the network can learn
- d None of the above

UNIT - III

3 Bayesian Learning, Computational Learning and Instance Based Learning

3.1 : Bayesian Learning and Bayes Theorem

Q.1 What is Bayesian neural network ?

Ans. : Bayesian Neural Network (BNN) refers to extending standard networks with posterior inference. Standard NN training via optimization is equivalent to Maximum Likelihood Estimation (MLE) for the weights.

Q.2 What are the features of Bayesian learning methods ?

Ans. : 1. Each observed training example can incrementally decrease or increase the estimated probability that a hypothesis is correct.

2. Prior knowledge can be combined with observed data to determine the final probability of a hypothesis.
3. Bayesian methods can accommodate hypotheses that make probabilistic predictions.
4. New instances can be classified by combining the predictions of multiple hypotheses, weighted by their probabilities.
5. Even in cases where Bayesian methods prove computationally intractable, they can provide a standard of optimal decision making against which other practical methods can be measured.

Q.3 What is the practical difficulty in applying Bayesian methods ?

Ans. : Practical difficulty in applying Bayesian methods are as follows :

1. Require initial knowledge of many probabilities. When these probabilities are not known in advance they are often estimated based on background knowledge, previously available data, and assumptions about the form of the underlying distributions.
2. The significant computational cost required to determine the Bayes optimal hypothesis in the general case.

Q.4 What is Bayes theorem ? How to select Hypotheses ?

Ans. : • In machine learning, we try to determine the best hypothesis from some hypothesis space H, given the observed training data D.

- In Bayesian learning, the best hypothesis means the most probable hypothesis, given the data D plus any initial knowledge about the prior probabilities of the various hypotheses in H.
- Bayes theorem provides a way to calculate the probability of a hypothesis based on its prior probability, the probabilities of observing various data given the hypothesis, and the observed data itself.
- Bayes' theorem is a method to revise the probability of an event given additional information.
- Bayes's theorem calculates a conditional probability called a posterior or revised probability.
- Bayes' theorem is a result in probability theory that relates conditional probabilities. If A and B denote two events, P(A|B) denotes the conditional probability of A occurring, given that B occurs. The two conditional probabilities P(A|B) and P(B|A) are in general different.
- This theorem gives a relation between P(A|B) and P(B|A). An important application of Bayes' theorem is that it gives a rule how to update or revise the strengths of evidence-based beliefs in light of new evidence a posteriori.
- A prior probability is an initial probability value originally obtained before any additional information is obtained.

- A posterior probability is a probability value that has been revised by using additional information that is later obtained.
- If A and B are two random variables

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

- In the context of classifier hypothesis h and training data I.

$$p(h/I) = \frac{P(I/h)P(h)}{P(I)}$$

Where (h) = Prior probability of hypothesis h

(I) = Prior probability of training data I

(h|I) = Probability of h given I

P (I|h) = Probability of I given h

Choosing the Hypotheses

- Given the training data, we are interested in the most probable hypothesis. The learner considers some set of candidate hypotheses H and it is interested in finding the most probable hypothesis $h \in H$ given the observed data D.
- Any such maximally probable hypothesis is called a maximum a posteriori (MAP) hypothesis h_{MAP}
- Maximum a posteriori hypothesis (h_{MAP}).

$$h_{MAP} = \underset{h \in H}{\text{argmax}}\, P(h/I)$$

$$= \underset{h \in H}{\text{argmax}} \frac{P(I/h)P(h)}{P(I)}$$

$$= \underset{h \in H}{\text{argmax}}\, P(I/h)P(h)$$

- If every hypothesis is equally probable, $P(h_i) = P(h_j)$ for all h_i and h_j in H.
- P(I/h) is often called the likelihood of the data I given h. Any hypothesis that maximizes P(I/h) is called a maximum likelihood (ML) hypothesis, h_{ML}.

$$h_{ML} = \underset{h \in H}{\text{argmax}}\, P(I/h)$$

Q.5 At a certain university, 4% of men are over 6 feet tall and 1% of women are over 6 feet tall. The total student population is divided in the ratio 3:2 in favour of women. If a student is selected at random from among all those over six feet tall, what is the probability that the student is a woman ?

Ans. : Let us assume following :

M = (Student is Male),

F = (Student is Female),

T = (Student is over 6 feet tall).

Given data : P(M) = 2/5,

P(F) = 3/5,

P(T|M) = 4/100

P(T|F) = 1/100.

We require to find P(F|T) ?

Using Bayes' Theorem we have :

$$P(F/T) = \frac{P(T/F)\,P(F)}{P(T/F)\,P(F) + P(T/M)\,P(M)}$$

$$= \frac{\frac{1}{100} \times \frac{3}{5}}{\frac{1}{100} \times \frac{3}{5} + \frac{4}{100} \times \frac{2}{5}} = \frac{\frac{3}{500}}{\frac{3}{500} + \frac{8}{500}}$$

$$P(F/T) = \frac{3}{11}$$

Q.6 Bag contains 5 red balls and 2 white balls. Two balls are drawn successivly without replacement. Draw the probability tree for this.

Sol. : Let R_1 = for the event of getting a red ball on the first draw, W_2 for getting a white ball on the second draw, and so forth. Here's the probability tree.

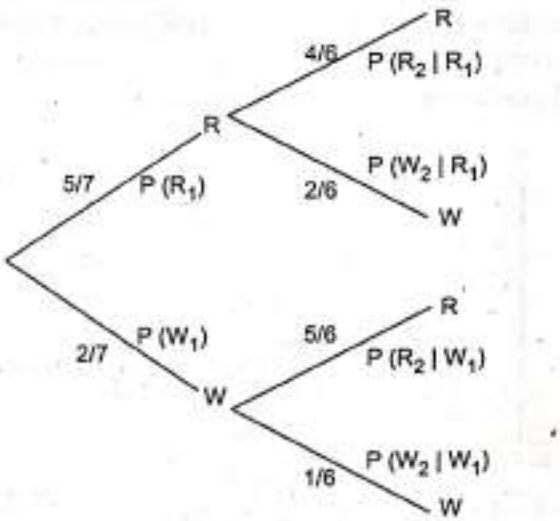


Fig. Q.6.1

3.2 : Maximum Likehood and Least Squared Error Hypotheses

Q.7 What do you mean by least square method ?

Ans. : Least squares is a statistical method used to determine a line of best fit by minimizing the sum of squares created by a mathematical function. A "square" is determined by squaring the distance between a data point and the regression line or mean value of the data set.

Q.8 What is maximum likelihood estimation ?

Ans. : Maximum-Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters.

Q.9 Briefly discuss least square method. List disadvantages of least square method.

Ans. :

- The method of least squares is about estimating parameters by minimizing the squared discrepancies between observed data, on the one hand, and their expected values on the other.
- Considering an arbitrary straight line, $y = b_0 + b_1 x$, is to be fitted through these data points. The question is "Which line is the most representative" ?
- What are the values of b_0 and b_1 such that the resulting line "best" fits the data points ? But, what goodness-of-fit criterion to use to determine among all possible combinations of b_0 and b_1 ?

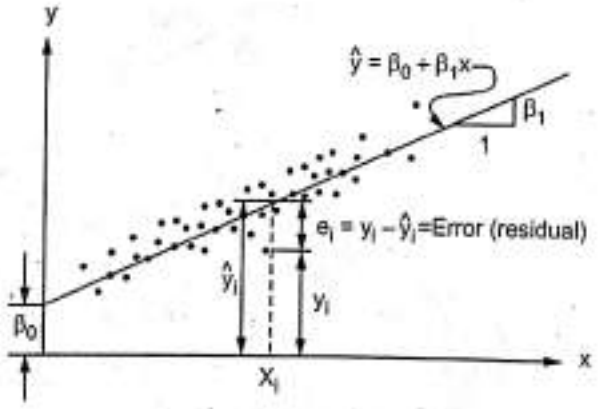


Fig. Q.9.1

- The Least Squares (LS) criterion states that the sum of the squares of errors is minimum. The least-squares solutions yields **y(x)** whose elements sum to 1, but do not ensure the outputs to be in the range [0,1].
- How to draw such a line based on data points observed ? Suppose a imaginary line of $y = a + bx$.
- Imagine a vertical distance between the line and a data point $E = Y - E(Y)$.
- This error is the deviation of the data point from the imaginary line, regression line. Then what is the best values of a and b ? A and b that minimizes the sum of such errors.

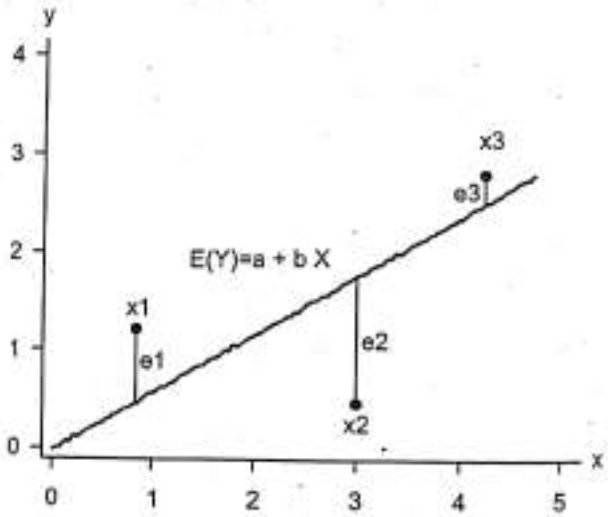


Fig. Q.9.2

- Deviation does not have good properties for computation. Then why do we use squares of deviation ? Let us get a and b that can minimize the sum of squared deviations rather than the sum of deviations. This method is called **least squares.**
- Least squares method minimizes the sum of squares of errors. Such a and b are called least squares estimators i.e. estimators of parameters α and β.
- The process of getting parameter estimators (e.g., a and b) is called **estimation.** Lest squares method is the estimation method of Ordinary Least Squares (OLS).

Disadvantages of least square

1. Lack robustness to outliers.
2. Certain datasets unsuitable for least squares classification.
3. Decision boundary corresponds to ML solution.

Q.10 Fit a straight line to the points in the table. Compute m and b by least squares.

Points	x	y
A	3.00	4.50
B	4.25	4.25
C	5.50	5.50
D	8.00	5.50

Ans. : Represent in matrix form :

$$\begin{bmatrix} 3.00 & 1 \\ 4.25 & 1 \\ 5.50 & 1 \\ 8.00 & 1 \end{bmatrix} \begin{bmatrix} m \\ b \end{bmatrix} = \begin{bmatrix} 4.50 \\ 4.25 \\ 5.50 \\ 5.50 \end{bmatrix} + \begin{bmatrix} v_A \\ v_B \\ v_C \\ v_D \end{bmatrix}$$

$$X = \begin{bmatrix} m \\ b \end{bmatrix} = (A^T A)^{-1} (A^T L)$$

$$= \begin{bmatrix} 121.3125 & 20.7500 \\ 20.7500 & 4.0000 \end{bmatrix}^{-1} \begin{bmatrix} 105.8125 \\ 19.7500 \end{bmatrix}$$

$$= \begin{bmatrix} 0.246 \\ 3.663 \end{bmatrix}$$

$$V = AX - L$$

$$= \begin{bmatrix} 3.00 & 1 \\ 4.25 & 1 \\ 5.50 & 1 \\ 8.00 & 1 \end{bmatrix} \begin{bmatrix} 0.246 \\ 3.663 \end{bmatrix} - \begin{bmatrix} 4.50 \\ 4.25 \\ 5.50 \\ 5.50 \end{bmatrix} = \begin{bmatrix} -0.10 \\ 0.46 \\ -0.48 \\ 0.13 \end{bmatrix}$$

Q.11 Explain with example maximum likelihood estimation.

Ans. : • Maximum-Likelihood Estimation (MLE) is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, maximum-likelihood estimation provides estimates for the model's parameters.

$X_1, X_2, X_3, \ldots, X_n$ have joint density denoted

$f_\theta(x_1, x_2, \ldots, x_n) = f(x_1, x_2, \ldots, x_n|\theta)$

Given observed values $X_1 = x_1, x_2 = x_2, \ldots, X_n = x_n$, the likelihood of θ is the function

$$\text{lik}(\theta) = f(x_1, x_2, \ldots, x_n|\theta)$$

Considered as a function of θ.

• If the distribution is discrete, f will be the frequency distribution function.

• The maximum likelihood estimate of θ is that value of that maximises lik (θ) : It is the value that makes the observed data the most probable.

Examples of maximizing likelihood :

• A random variable with this distribution is a formalization of a coin toss. The value of the random variable is 1 with probability θ and 0 with probability $1 - \theta$. Let X be a Bernoulli random variable and let x be an outcome of X, then we have

$$P(X = x) = \begin{bmatrix} \theta & \text{if } x=1 \\ 1-\theta & \text{if } x=0 \end{bmatrix}$$

• Usually, we use the notation P(.) for a probability mass and the notation p(.) for a probability density. For mathematical convenience write P(X) as

$$P(X = x) = \theta^x (1-\theta)^{1-x}$$

Q.12 What is gradient search to maximize likelihood in a neural net ?

Ans. : • Develop a method for computers to "understand" speech using mathematical methods. For a D-dimensional input vector o, the Gaussian distribution with mean μ and positive definite covariance matrix $\sum$ can be expressed as

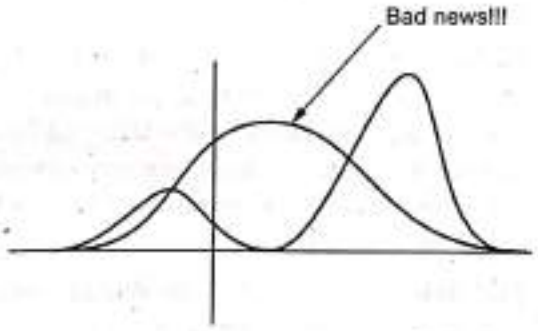


Fig. Q.12.1

$$N(o, \mu \sum) = \frac{1}{(2\pi)^{\frac{D}{2}} |\sum^{-1}|^{1/2}} e^{-1/2 (0-\mu)^T \sum^{-1} (0-\mu)}$$

• The distribution is completely described by the D parameters representing μ and the D(D + 1)/2 parameters representing the symmetric covariance matrix $\sum$.

• Single Gaussian may do a bad job of modeling distribution in any dimension :

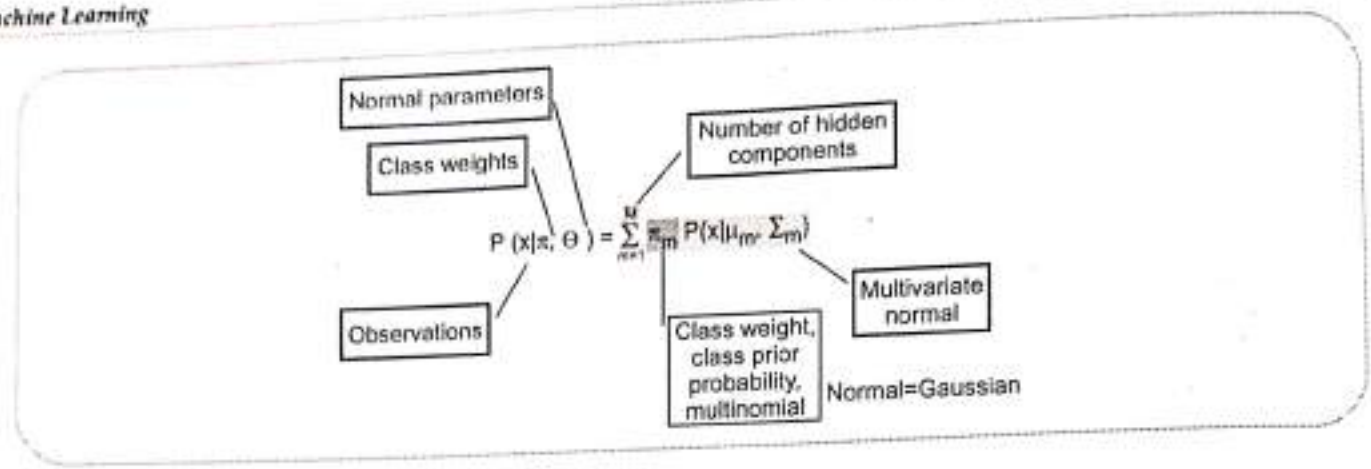


Fig. Q.12.2

- Solution : Mixtures of Gaussians is a solution for this problem.
- A formalism for modeling a probability density function as a sum of parameterized functions.

3.3 : Minimum Description Length Principle

Q.13 Explain minimum description length principle.

Ans. : • The Minimum Description Length (MDL) criteria in machine learning says that the best description of the data is given by the model which compresses it the best.

- Put another way, learning a model for the data or predicting it is about capturing the regularities in the data and any regularity in the data can be used to compress it. Thus, the more we can compress a data, the more we have learnt about it and the better we can predict it.
- The MDL principle states that one should prefer the model that yields the shortest description of the data when the complexity of the model itself is also accounted for.
- The Minimum Description Length principle is motivated by interpreting the definition of h_{MAP} in the light of basic concepts from information theory. Consider definition of h_{MAP}.

$$h_{MAP} = \underset{h \in H}{\text{argmax}}\ P(I/h)p(h)$$

- The MDL principle recommends choosing the hypothesis that minimizes the sum of these two description lengths.
- Assume the codes C_1 and C_2 to represent the hypothesis and the data given the hypothesis, we can state the MDL principle as

$$h_{MDL} = \underset{h \in H}{\text{argmin}}\ L_{C_1}(h) + L_{C_2}(D/h)$$

- The above analysis shows that if we choose C_1 to be the optimal encoding of hypotheses C_H and if we choose C_2 to be the optimal encoding $C_{I/h}$ then $h_{MDL} = h_{MAP}$.

3.4 : Bayes Optimal Classifier and Gibbs Algorithm

Q.14 Define Gibbs algorithm.

Ans. : The Gibbs algorithm defined as follows :

1. Choose a hypothesis **h** from H at random, according to the posterior probability distribution over H.
2. Use **h** to predict the classification of the next instance **x**.

Q.15 What is the Bayes optimal classifier ?

Ans. : • Bayes classifier is a classifier that minimizes the error in a probabilistic manner. If it is Bayes optimal, then the errors are weighed using the joint probability distribution between the input and the output sets.

- The Bayes error is then the error of the Bayes classifier.

3.5 : Naïve Bayes Classifier

Q.16 What is Naïve Bayes Classifiers ?

Ans. : • Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. It is highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem.

- A Naive Bayes Classifier is a program which predicts a class value given a set of attributes.
- For each known class value,
  1. Calculate probabilities for each attribute, conditional on the class value.
  2. Use the product rule to obtain a joint conditional probability for the attributes.
  3. Use Bayes rule to derive conditional probabilities for the class variable.
- Once this has been done for all class values, output the class with the highest probability.
- Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
- The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class.

3.6 : Bayesian Belief Networks

Q.17 Describe Bayesian belief network.

Ans. : Bayesian belief network describes the probability distribution governing a set of variables by specifying a set of conditional independence assumptions along with a set of conditional probabilities.

Q.18 Explain with example how Bayesian belief network is represented ?

Ans. :

- Bayesian belief networks represent the full joint distribution over the variables more compactly with a smaller number of parameters.
- It take advantage of conditional and marginal independences among random variables
- A and B are independent then $P(A, B) = P(A)P(B)$
- A and B are conditionally independent given C

$P(A, B \mid C) = P(A \mid C)P(B \mid C)$

$P(A \mid C, B) = P(A \mid C)$

- **Example : Alarm system example.**
- Assume your house has an alarm system against burglary. You live in the seismically active area and the alarm system can get occasionally set off by an earthquake.
- You have two neighbors, Mary and John, who do not know each other. If they hear the alarm they call you, but this is not guaranteed.
- We want to represent the probability distribution of events : Burglary, Earthquake, Alarm, Mary calls and John calls

Causal relations :

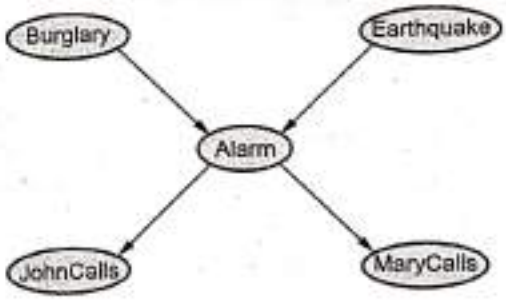


Fig. Q.18.1

Directed acyclic graph :

- Nodes = Random variables

Burglary, Earthquake, Alarm, Mary calls and John calls

- Links = Direct (causal) dependencies between variables.

The chance of Alarm is influenced by Earthquake, The chance of John calling is affected by the Alarm.

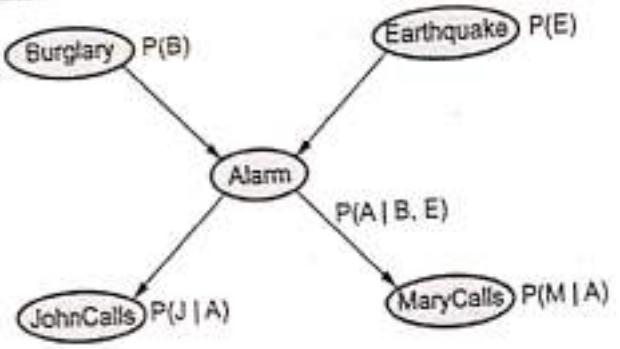


Fig. Q.18.2

Local conditional distributions : Relate variables and their parents

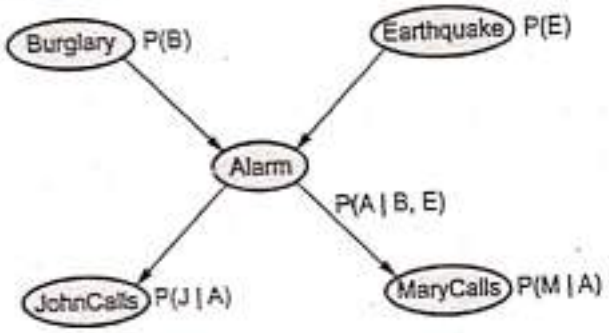


Fig. Q.18.3

Bayesian belief network :

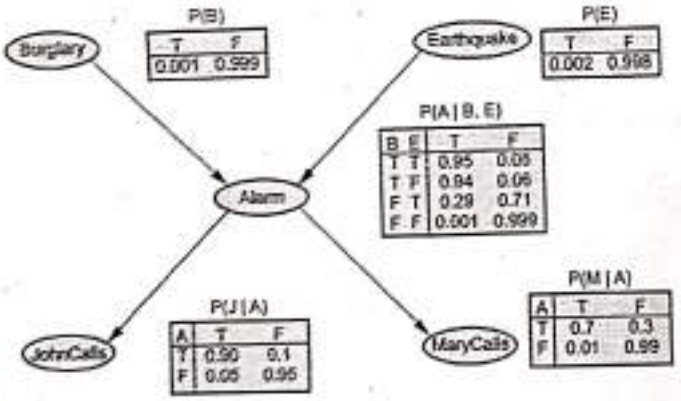


Fig. Q.18.4

3.7 : The EM Algorithm

Q.19 Write short note on EM algorithm.

Ans. : • Expectation-Maximization (EM) is an iterative method used to find maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved, also called latent, variables.

- EM alternates between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found in the E step.
- The parameters found on the M step are then used to start another E step, and the process is repeated until some criterion is satisfied. EM is frequently used for data clustering like for example in Gaussian mixtures.
- In the **Expectation step**, find the expected values of the latent variables (here you need to use the current parameter values).
- In the **Maximization step**, first plug in the expected values of the latent variables in the log-likelihood of the augmented data. Then maximize this log-likelihood to reevaluate the parameters.
- Expectation-Maximization (EM) is a technique used in point estimation. Given a set of observable variables X and unknown (latent) variables Z we want to estimate parameters θ in a model.
- The expectation maximization (EM) algorithm is a widely used maximum likeli-hood estimation procedure for statistical models when the values of some of the variables in the model are not observed
- The EM algorithm is an elegant and powerful method for finding the maximum likelihood of models with hidden variables. The key concept in the EM algorithm is that it iterates between the expectation step (E-step) and maximization step (M-step) until convergence.
- In the E-step, the algorithm estimates the posterior distribution of the hidden variables Q given the observed data and the current parameter settings and in the M-step the algorithm calculates the ML parameter settings with Q fixed.
- At the end of each iteration the lower bound on the likelihood is optimized for the given parameter setting (M-step) and the likelihood is set to that bound (E-step), which guarantees an increase in the likelihood and convergence to a local maximum, or global maximum if the likelihood function is unimodal.

- Generally, EM works best when the fraction of missing information is small and the dimensionality of the data is not too large. EM can require many iterations, and higher dimensionality can dramatically slow down the E-step.
- EM is useful for several reasons: conceptual simplicity, ease of implementation, and the fact that each iteration improves $l(\theta)$. The rate of convergence on the first few steps is typically quite good, but can become excruciatingly slow as you approach local optima.
- Sometimes the M-step is a constrained maximization, which means that there are constraints on valid solutions not encoded in the function itself.
- Expectation maximization is an effective technique that is often used in data analysis to manage missing data. Indeed, expectation maximization overcomes some of the limitations of other techniques, such as mean substitution or regression substitution. These alternative techniques generate biased estimates-and, specifically, underestimate the standard errors. Expectation maximization overcomes this problem.

3.8 : Introduction of Computational Learning Theory

Q.20 What is computational learning theory ?

Ans. : • Computational learning theory provides a formal framework in which to precisely formulate and address questions regarding the performance of different learning algorithms so that careful comparisons of both the predictive power and the computational efficiency of alternative learning algorithms can be made.

- Three key aspects that must be formalized are the way in which the learner interacts with its environment the definition of successfully completing the learning task and a formal definition of efficiency of both data usage (sample complexity) and processing time (time complexity)

3.9 : Probably Learning an Approximately Correct Hypothesis

Q.21 Define Probably Approximately Correct Learning.

Ans. : A concept class C is said to be PAC learnable using a hypothesis class H if there exists a learning algorithm L such that for all concepts in C, for all instance distributions D on an instance space X, $\forall \varepsilon$, $\delta (0 < \varepsilon, \delta < 1)$, L, when given access to the Example oracle, produces, with probability at least $(1 - \delta)$, a hypothesis h from H with error no more than ε.

Q.22 Define consistent learner.

Ans. : A consistent learner is one that returns some hypothesis h from the hypothesis class H that is consistent with a random sequence of m examples. A consistent learner is a MAP learner, if all hypotheses are a-priori equally likely.

Q.23 Discuss briefly Probably Approximately Correct Learning.

Ans. : • PAC is a nice formalism for deciding how much data you need to collect in order for a given classifier to achieve a given probability of correct predictions on a given fraction of future test data.

- To understand what this model is all about, it's probably easiest just to give an example. Say there's a hidden line on the chalk board.
- Given a point on the board, we need to classify whether it's above or below the line. To help, we'll get some sample data, which consists of random points on the board and whether each point is above or below the line.
- After seeing, say, twenty points, you won't know exactly where the line is, but you'll probably know roughly where it is. And using that knowledge, you'll be able to predict whether most future points lie above or below the line.
- Suppose we have agreed that predicting the right answer "most of the time" is okay. Is any random choice of twenty points going to give you that ability ? No, because you could get really unlucky with the sample data, and it could tell you almost nothing about where the line is. Hence the "Probably" in PAC.

- X is the set of all possible examples. D is the distribution from which the examples are drawn
- H is the set of all possible hypotheses, $c \in H$
- m is the number of training examples. Then

error(h) = Pr(h(x) ≠ c(x) | x is drawn from X with D)

where h is approximately correct if error(h) ≤ε

- *Hypothesis h(X) is consistent with m examples and has an error of at most with probability* $1 - \delta$ This is a worst-case analysis. Note that the result is independent of the distribution D.
- **Curse of dimensionality :** If the number of features d is large, the number of samples n, may be too small for accurate parameter estimation.
- For accurate estimation, n should be much bigger than d^2, otherwise model is too complicated for the data, overfitting.

3.10 : Sample Complexity for Infinite Hypothesis Spaces

Q.24 Explain VC dimension.

Ans. : Vapnik-Chervonenkis (VC) dimension provides a measure of the complexity of a space of functions, and which allows the probably approximately correct framework to be extended to spaces containing an infinite number of functions.

- The Vapnik-Chervonenkis dimension is a measure of the complexity or capacity of a class of functions f(α). The VC dimension measures the largest number of examples that can be explained by the family f(α).
- The Vapnik-Chervonenkis dimension, VC(H), of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H. If arbitrarily large finite sets of X can be shattered by H, then VC(H) = ∞.
- The basic argument is that high capacity and generalization properties are at odds :

1. If the family f(α) has enough capacity to explain every possible dataset, we should not expect these functions to generalize very well.
2. On the other hand, if functions f(α) have small capacity but they are able to explain one particular dataset, we have stronger reasons to believe that they will also work well on unseen data.

- **Shattering a set of examples :** Assume a binary classification problem with N examples in R^D and consider the set of $2^{|N|}$ possible dichotomies. For instance, with N = 3 examples, the set of all possible dichotomies is **{(000), (001), (010), (011), (100), (101), (110), (111)}**. A class of functions f(α) is said to shatter the dataset if, for every possible dichotomy, there is a function in f(α) that models it.
- Consider as an example a finite concept class $C = \{c_1, ..., c_4\}$ applied to three instance vectors with the results :

	X_1	X_2	X_3
c_1	1	1	1
c_2	0	1	1
c_3	1	0	0
c_4	0	0	0

Then :

$\pi_c(\{X_1\}) = \{(0), (1)\}$ shattered

$\pi_c(\{X_1, X_3\}) = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ shattered

$\pi_c(\{X_2, X_3\}) = \{(0, 0), (1, 1)\}$ not shattered

- The **VC dimension VC(f)** is the size of the largest dataset that can be shattered by the set of functions f (α). If the VC dimension of (α) is h, then there exists at least one set of h points that can be shattered by (α), but in general it will not be true that every set of h points can be shattered.
- **Example of VCdim : axis aligned rectangles**
- If we had five points, then at most four of the points determine the minimal rectangle that contains the whole set. Then no rectangle is consistent with the labeling that assigns these four boundary points "+" and assigned the remaining point a "-". Therefore,

VCdim (axis-aligned rectangles ε R^2) = 4

- The VC dimension cannot be accurately estimated for non-linear models such as neural networks. The

VC dimension may be infinite requiring infinite amount of data.

3.11 : The Mistake Bound Model of Learning

Q.25 Define mistake bound model.

Ans. : Algorithm A has mistake-bound M for learning class C if A makes at most M mistakes on any sequence that is consistent with a function in C.

Q.26 Explain Weighted Majority Algorithm.

Ans. :

- A classifier combination method
- Takes a weighted vote among a pool of prediction algorithms, e.g., alternative hypotheses in H, or alternative learning algorithms
- It begins by weighting each algorithm by 1
- Whenever an algorithm misclassifies its weight is decreased by β, where $0 < \beta < 1$.
- If A is any set of n prediction algorithms.
- If k is the minimum number of mistakes made by any algorithm in A.
- The number of mistakes made over any training sequence is at most.

3.12 : Introduction of Instance-based Learning Methods

Q.27 What is instance-based learning ? List its advantages and disadvantages.

Ans. : • All learning methods presented so far construct a general explicit description of the target function when examples are provided.

- Instance-based learning methods simply store the training examples instead of learning explicit description of the target function.
- Generalizing the examples is postponed until a new instance must be classified.
- When a new instance is encountered, its relationship to the stored examples is examined in order to assign a target function value for the new instance.
- Instance-based learning includes nearest neighbor, locally weighted regression and case-based reasoning methods.
- Instance-based methods are sometimes referred to as lazy learning methods because they delay processing until a new instance must be classified.
- A key advantage of lazy learning is that instead of estimating the target function once for the entire instance space, these methods can estimate it locally and differently for each new instance to be classified.

Advantages :

1. Instead of estimating for the whole instance space, local approximations to the target function are possible
2. Especially if target function is complex but still decomposable.

Disadvantages :

1. Classification costs are high.
2. Typically all attributes are considered when attempting to retrieve similar training examples.

3.13 : k-Nearest Neighbour Learning

Q.28 What is K-Nearest Neighbour Methods ?

Ans. : • The K-nearest neighbor (KNN) is a classical classification method and requires no training effort, critically depends on the quality of the distance measures among examples.

- The KNN classifier uses Mahalanobis distance function. A sample is classified according to the majority vote of its nearest K training samples in the feature space. Distance of a sample to its neighbors is defined using a distance function

Q.29 What is Euclidean distance ?

Ans. : • The Euclidean distance is the most common distance metric used in low dimensional data sets. It is also known as the L_2 norm.

- The Euclidean distance is the usual manner in which distance is measured in real world.

Q.30 Define Mahalanobis distance.

Ans. : Mahalanobis distance is also called quadratic distance.

- Mahalanobis distance is a distance measure between two points in the space defined by two or more correlated variables. Mahalanobis distance takes the correlations within a data set between the variableinto considerations.
- While Euclidean metric is useful in low dimensions, it doesn't work well in high dimensions and for categorical variables.
- The drawback of Euclidean distance is that it ignores the similarity between attributes. Each attribute is treated as totally different from all of the attributes.

Q.31 List out the steps that need to be carried out during the KNN algorithm.

Ans. : Steps are as follows :

a. Divide the data into training and test data.
b. Select a value K.
c. Determine which distance function is to be used.
d. Choose a sample from the test data that needs to be classified and compute the distance to its n training samples
e. Sort the distances obtained and take the k-nearest data samples.
f. Assign the test class to the class based on the majority vote of its K neighbors

Q.32 What are the advantages and disadvantages of KNN ?

Ans. : Advantages

1. The KNN algorithm is very easy to implement.
2. Nearly optimal in the large sample limit.
3. Uses local information, which can yield highly adaptive behavior.
4. Lends itself very easily to parallel implementations.

Disadvantages

1. Large storage requirements.
2. Computationally intensive recall.
3. Highly susceptible to the curse of dimensionality.

Q.33 Which are the performance factors that influence KNN algorithm ?

Ans. : The performance of the KNN algorithm is influenced by three main factors :

1. The distance function or distance metric used to determine the nearest neighbors.
2. The decision rule used to derive a classification from the K-nearest neighbors.
3. The number of neighbors used to classify the new example.

3.14 : Locally Weighted Regression

Q.34 Write short note on Locally Weighted Regression.

Ans. : • KNN forms local approximation to f for each query point xq.

- Why not form an explicit approximation f(x) for region surrounding xq? Use Locally Weighted Regression.
- Locally Weighted Regression (LWR) is a memory-based method that performs a regression around a point of interest using only training data that are ``local" to that point.
- Locally weighted regression uses nearby or distance-weighted training examples to form this local approximation to f.
- We might approximate the target function in the neighborhood surrounding x, using a linear function, a quadratic function, a multilayer neural network.
- The phrase "locally weighted regression" is called
  a. local because the function is approximated based only on data near the query point,
  b. weighted because the contribution of each training example is weighted by its distance from the query point, and
  c. regression because this is the term used widely in the statistical learning community for the problem of approximating real-valued functions.
- Given a new query instance xq, the general approach in locally weighted regression is to construct an approximation f that fits the training examples in the neighborhood surrounding xq.
- This approximation is then used to calculate the value f(xq), which is output as the estimated target value for the query instance.

Locally Weighted Linear Regression

- Let us consider the case of locally weighted regression in which the target function f is approximated near x, using a linear function of the form

$$\hat{f}(x) = w_0 + w_1 a_1(x) + \ldots + w_n a_n(x)$$

Where $a_i(x)$ denotes the value of the ith attribute of the instance x.

- Minimize the squared error :

$$E_3(x_q) = \frac{1}{2} \sum_{x \in \text{k nearest nbrs of } x_q} (f(x) - \hat{f}(x)^2 K(d(x_q, x))$$

- Kernel function K is the function of distance that is used to determine the weight of each training example

$$\Delta w_j = \eta \sum_{x \in \text{ k nearest nbrs of } x_q} K(d(x_q, x)) - (f(x) - (\hat{f}(x)) a_j(x)$$

3.15 : Radial Basis Functions

Q.35 What is radial basis function network ?

Ans. : • Radial Basis Function (RBF) network is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters.

- RBF networks form a special class of neural networks, which consist of three layers.
- The input layer is used only to connect the network to its environment.
- The hidden layer contains a number of nodes, which apply a nonlinear transformation to the input variables, using a radial basis function, such as the Gaussian function, the thin plate spline function etc.
- The output layer is linear and serves as a summation unit.

Q.36 List the features of Radial Basis Function (RBF) networks.

Ans. : Features are as follows :

1. They are two-layer feed-forward networks.
2. The hidden nodes implement a set of radial basis functions (e.g. Gaussian functions).
3. The output nodes implement linear summation functions as in an MLP.
4. The network training is divided into two stages : first the weights from the input to hidden layer are determined, and then the weights from the hidden to output layer.
5. The training/learning is very fast.
6. The networks are very good at interpolation

Q.37 What is a radial basis function network ? Explain with architecture.

Ans. : • Radial Basis Function Networks (RBFN) are a variant of the three-layer feedforward neural networks. They contain a pass-through input layer, a hidden layer and an output layer.

- Fig. Q.37.1 shows a schematic diagram of an RBFN.

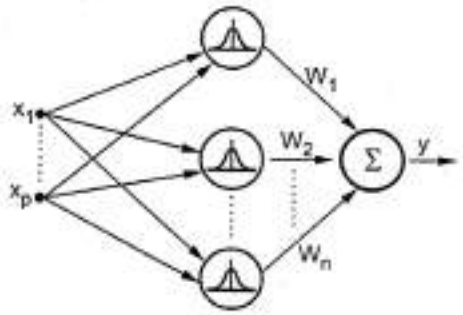


Fig. Q.37.1 Schematic diagram of RBFN

- The transfer function in the hidden layer is called a radial basis function (RBF). The RBF networks divide the input space into hyperspheres, and utilize a special kind of neuron transfer function.
- Radial basis functions are frequently used to create neural networks for regression-type problems. Their characteristic feature is that their response decreases (or increases) monotonically with distance from a central point.
- The hidden unit is given as :

$$W_i = R_i(X) = R_i(\|x - u_i / \sigma_i\) \quad i = 1, 2, \ldots H.$$

Where x = Multi-dimensional input vector

u_i = Vector with the same dimension as x

H = Number of radial basic function

$R_i(.)$ = It is i^{th} radial basis function with a single maximum at the origin.

- There are no connection weights between the input layer and the hidden layer.
- Typically $R_i(.)$ is a Gaussian function.

$$R_i(x) = \exp\left(-\frac{\|x-u_i\|^2}{2\sigma_i^2}\right)$$

Or a logistic function

$$R_i(x) = \frac{1}{1+\exp[\|x-u_i\|^2/\sigma_i^2]}$$

- The output of the RBFN can be computed in two ways :

1. Weight sum of the output value associated with each receptive field :

$$d(x) = \sum_{i=1}^{H} c_i w_i = \sum_{i=1}^{H} c_i R_i(X)$$

where c_i = output value associated with the ith receptive field.

2. Weighted average of the associated with each receptive field :

$$d(x) = \frac{\sum_{i=1}^{H} c_i w_i}{\sum_{i=1}^{H} w_i}$$

$$= \frac{\sum_{i=1}^{H} c_i R_i(x)}{\sum_{i=1}^{H} R_i(x)}$$

- Moody darken RBFN may be extended by assigning a linear function to the output function of each receptive field i.e. making c_i a linear combination of the input variables plus a constant :

$$c_i = a_i^T x + b_i$$

Where a_i is a parameter vector and b_i is a scalar parameter.

- A hidden neuron is more sensitive to data points near its center. For Gaussian RBF the sensitively may be turned by adjusting the spread σ, where a larger spread implies less sensitivity.
- An RBFN approximation capacity may be further improved with supervised adjustments of the center and shape of the receptive field function.

Q.38 Compare RBF network with multilayer perceptron.

Ans. :

No.	RBF networks	Multilayer perceptrons
1	An RBFN has a single hidden layer.	MLP may have one or more hidden layers.
2	Hidden layer is nonlinear and output layer is linear.	Hidden and output layer used as pattern classifier are usually nonlinear.
3	The argument of the activation function of each hidden unit computes the Eucliden norm between the input vector and the centre of that unit.	The activation function of each hidden unit computes the inner product of the input vector and the synaptic weight vector of that unit.
4	RBF networks using exponentially decaying localized nonlinearities construct local approximations to nonlinear input-output mappings.	MLPs construct global approximations to nonlinear input-output mapping.
5	Computation nodes in the hidden layer of an RBF network are quite different and serve a different purpose from those in the output layer of the network.	Computation nodes of an MLP, located in a hidden or an output layer, share a common neuronal model.

3.16 : Case-based Reasoning

Q.39 What is case-based reasoning ? Explain its steps.

Ans. : • Case based reasoning stores information from previous experiences. Using previously gained knowledge to solve current problems. It is similar to human problem solving methods.

- CBR has been applied to problems such as conceptual design of mechanical devices based on a stored library of previous designs.
- Basic Steps :
  1. Identify the problem/case.
  2. Look for a similar, previously experienced case.
  3. Predict a solution, possibly different from past experiences.
  4. Evaluate the solution.
  5. Update the system with the results.

- Case-based reasoning can be used for classification and regression. It is also applicable when the cases are complicated, such as in legal cases, where the cases are complex legal rulings, and in planning, where the cases are previous solutions to complex problems.
- A common example of a case-based reasoning system is a helpdesk that users call with problems to be solved.
- For example, case-based reasoning could be used by the diagnostic assistant to help users diagnose problems on their computer systems.
- When a user gives a description of their problem, the closest cases in the case base are retrieved.
- The diagnostic assistant can recommend some of these to the user, adapting each case to the user's particular situation.
- An example of adaptation is to change the recommendation based on what software the user has, what method they use to connect to the Internet, and the brand of printer.
- If one of the cases suggested works, that can be recorded in the case base to make that case be more important when another user asks a similar question.
- If none of the cases found works, some other problem solving can be done to solve the problem, perhaps by adapting other cases or having a human help diagnose the problem.
- When the problem is finally fixed, what worked in that case can be added to the case base.

3.17 : Remarks on Lazy and Eager Techniques

Q.40 Discuss concept of weak and eager learner.

Ans. : • Eager learning is a learning method in which the system tries to construct a general, input-independent target function during training of the system, as opposed to lazy learning, where generalization beyond the training data is delayed until a query is made to the system.

• Combining several weak learners to give a strong learner. It is a kind of multiclassifier systems and meta-learners. Ensemble typically applied to a single type of weak learner.

- **Lazy learning** (e.g., instance-based learning) : Simply stores training data (or only minor processing) and waits until it is given a test tuple.
- **Eager learning** (the above discussed methods) : Given a set of training tuples, constructs a classification model before receiving new (e.g., test) data to classify.
- **Lazy** : less time in training but more time in predicting.
- Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form an implicit global approximation to the target function.
- **Eager** : must commit to a single hypothesis that covers the entire instance space.

Fill in the Blanks for Mid Term Exam

Q.1 _____ networks are a type of artificial neural network constructed from spatially localized kernel functions.

Q.2 A _____ estimate of a parameter consists of an interval of numbers along with a probability that the interval contains the unknown parameter.

Q.3 Bayes theorem provides a way to calculate the probability of a hypothesis based on its _____, the probabilities of observing various data given the hypothesis, and the observed data itself.

Q.4 The Minimum Description Length principle recommends choosing the _____ that minimizes the sum of e two description lengths.

Q.5 Given a new instance to classify, the _____ algorithm simply applies a hypothesis drawn at random according to the current posterior probability distribution.

Q.6 The EM algorithm has been used to train Bayesian belief networks as well as _____.

Q.7 PAC-learnability is largely determined by the number of training examples required by the _____.

UNIT - IV

4 Genetic Algorithm

4.1 : Genetic Algorithm : Motivation

Q.1 What is Genetic algorithm ?

Ans. : A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems.

Q.2 List the factors motivated the popularity of genetic algorithms. ☞ **[JNTU : Dec-17, Marks 3]**

Ans. : Genetic algorithm is that the problem solving strategy involves using "the strings' fitness to direct the search; therefore they do not require any problem-specific knowledge of the search space, and they can operate well on search spaces that have gaps, jumps, or noise. As each individual string within a population directs the search, the genetic algorithm searches, in parallel, numerous points on the problem state space with numerous search directions.

Q.3 Give an example for fitness function in genetic algorithms. ☞ **[JNTU : Dec-16, Marks 3]**

Ans. : • A fitness function is a particular type of objective function that is used to summaries, as a single figure of merit, how close a given design solution is to achieving the set aims.

- Fitness functions are used in genetic programming and genetic algorithms to guide simulations towards optimal design solutions.

Q.4 Explain the "Darwinian theory of survival"

Ans. : More individuals are produced each generation that can survive. Phenotypic variation exists among individuals and the variation is heritable. Those individuals with heritable traits better suited to the environment will survive.

Q.5 List the basic components used in all genetic algorithms.

Ans. : The basic components common to almost all genetic algorithms are :

1. Fitness function for optimization
2. A population of chromosomes
3. Selection of which chromosomes will reproduce
4. Crossover to produce next generation of chromosomes
5. Random mutation of chromosomes in new generation

Q.6 Compare and contract genetic algorithm with traditional algorithm.

Ans. :

Genetic algorithm	Traditional algorithm
GA generates a population of points at each iteration. The best point in the population approaches an optimal solution.	It generates a single point at each iteration. The sequence of points approaches an optimal solution.
Selects the next population by computation which uses random number generators	Selects the next point in the sequence by a deterministic computation.
Convergence in each iteration in problem independent	Improvement in each iteration is problem specific
Rules are probabilistic.	Rules are fully deterministic.

Q.7 What is use of crossover operator ?

Ans. : A crossover operator is used to recombine two strings to get a better string. In crossover operation, recombination process creates different individuals in the successive generations by

combining material from two individuals of the previous generation

Q.8 Explain two point crossover.

Ans. : Two crossover points are selected, binary string from the beginning of the chromosome to the first crossover point is copied from the first parent, the part from the first to the second crossover point is copied from the other parent and the rest is copied from the first parent again.

Q.9 What is crowding ?

Ans. : Crowding is a phenomenon in which some individual that is more highly fit than others in the population quickly reproduces, so that copies of this individual and very similar individuals take over a large fraction of the population.

Q.10 Why use Genetic algorithm ?

Ans. : • Genetic algorithms evaluate the target function to be optimized at some randomly selected points of the definition domain. Genetic algorithms can be used when no information is available about the gradient of the function at the evaluated points. The function itself does not need to be continuous or differentiable.

- Genetic algorithms can still achieve good results even in cases in which the function has several local minima or maxima.
- Genetic algorithms are stochastic search algorithms which act on a population of possible solutions. They are loosely based on the mechanics of population genetics and selection.
- Genetic Algorithms allow you to explore a space of parameters to find solutions that score well according to a "fitness function".
- Genetic Programming takes genetic algorithms a step further, and treats programs as the parameters. For example, you would breeding path finding algorithms instead of paths, and your fitness function would rate each algorithm based on how well it does.

Q.11 What are limitations on genetic algorithms ?

Ans. : • GAs are not guaranteed to find the global optimum solution to a problem.

- GAs are an extremely general tool and they have no specific way for solving particular problems.
- GAs are usually used when everything else is failed or when we don't have enough knowledge of the search space.
- Even when such specialized techniques exist, it is often interesting to hybridise them with a GA in order to possibly gain some improvements.

Q.12 Explain advantages of Genetic algorithm.

Ans. : • Genetic Algorithm is a stochastic algorithm.

- Randomness as an essential role in both selection and reproduction phases.
- Genetic algorithms always consider a population of solutions. A population base algorithm is also very amenable for parallelization.
- There is no particular requirement on the problem before using genetic algorithms, so it can be applied to resolve any problem (optimization).
- GAs are a new field and parts of the theory have still to be properly established. We can find almost as many opinions on GAs as there are researchers in this field.

Q.13 What is Fitness function ?

Ans. : • Fitness is an important concept in genetic algorithms. The fitness of a chromosome determines how likely it is that it will reproduce. Fitness is usually measured in terms of how well the chromosome solves some goal problem. Fitness can also be subjective (aesthetic). E.g., if the genetic algorithm is to be used to sort numbers, then the fitness of a chromosome will be determined by how close to a correct sorting it produces.

- A fitness function quantifies the optimality of a solution (chromosome) so that particular solution may be ranked against all the other solutions. A fitness value is assigned to each solution depending on how close it actually is to solving the problem.
- Ideal fitness function correlates closely to goal plus quickly computable.
- Example : In TSP, f(x) is sum of distances between the cities in solution. The lesser the value, the fitter the solution is.

- The performance of the individual strings is measured by a fitness function. A fitness function is a problem specific user defined heuristic. After each iteration, the members are given a performance measure derived from the fitness function and the "fittest" members of the population will propagate the next iteration.

$$\text{Fitness} = F_i,\ \text{Hit} = \lambda_i,\ \text{Survival} = \phi_i,\ \text{Death} = \delta_i$$

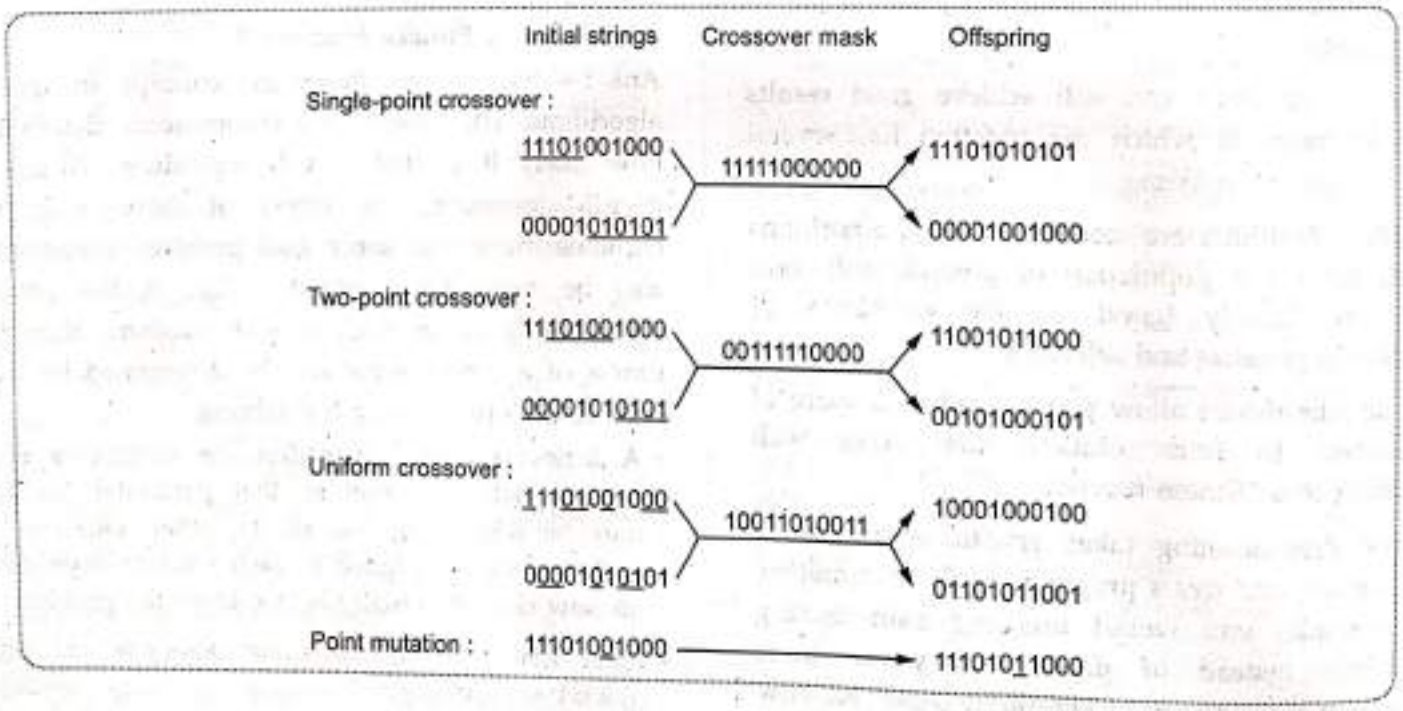
$$F_i = 2\lambda_i - \delta_i + \sum \phi_i$$

- The fitness function is defined over the genetic representation and measures the *quality* of the represented solution. The fitness function is always problem dependent.
- For instance, in the knapsack problem we want to maximize the total value of objects that we can put in a knapsack of some fixed capacity. A representation of a solution might be an array of bits, where each bit represents a different object and the value of the bit (0 or 1) represents whether or not the object is in the knapsack.
- Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The *fitness* of the solution is the sum of values of all objects in the knapsack if the representation is valid or 0 otherwise. In some problems, it is hard or even impossible to define the fitness expression; in these cases, interactive genetic algorithms are used.

Q.14 Consider the two strings as initial population for genetic algorithm and generate all possible off springs using various operators.
String 1: 11101001000
String 2: 00001010101

[JNTU : Dec-17, Marks 10]

Ans. :



4.2 : Genetic Programming

Q.15 What do you mean genetic programming ?

Ans. : Genetic programming is genetic algorithm wherein the population contains programs rather than bit strings.

Q.16 What is genetic programming ?

Ans. : • Genetic Programming (GP) is a method to evolve computer programs.

- Genetic programming is a model of programming which uses the ideas of biological evolution to handle a complex problem.
- Genetic programming can be viewed as an extension of the genetic algorithm, a model for testing and selecting the best choice among a set of results, each represented by a string.
- Genetic programming is a recent development in the area of evolutionary computation. It was greatly stimulated in the 1990s by **John Koza.**
- According to Koza, genetic programming searches the space of possible computer programs for a program that is highly fit for solving the problem at hand.
- A parse tree is a good representation of a computer program for Genetic Programming.

Q.17 Explain how genetic algorithms are different from evolutionary programming.

Ans. : 1. Genetic algorithms a coded form of the function values i.e. parameter set, rather than with the actual values themselves. For example, if we want to find the minimum of the function $f(x) = x^3 + x^2 + 5$, the GA would not deal directly with x or y values, but with strings that encode these values. For this case, strings representing the binary x values should be used.

2. Genetic algorithms use a set or population of points to conduct a search, not just a single point on the problem space. This gives GAs the power to search noisy spaces littered with local optimum points. Instead of relying on a single point to search through the space, the GAs looks at many different areas of the problem space at once, and uses all of this information to guide it.

3. Genetic algorithms use only payoff information to guide themselves through the problem space. Many search techniques need a variety of information to guide themselves. Hill climbing methods require derivatives, for example. The only information a GA needs is some measure of fitness about a point in the space. Once the GA knows the current measure of "goodness" about a point, it can use this to continue searching for the optimum.

4. GAs are probabilistic in nature, not deterministic. This is a direct result of the randomization techniques used by GAs.

5. GA is inherently parallel. Here lies one of the most powerful features of genetic algorithms. GA's, by their nature, are very parallel, dealing with a large number of points simultaneously.

Parameters	Genetic Algorithms	Traditional Methods
Work with	Coding of parameter set	Parameters directly
Use information	Payoff i.e. objective function	Payoff plus derivatives etc.
Rules	Probabilistic	Fully deterministic
Search	A population of points	A population of points a single point

4.3 : Models of Evolution and Learning

Q.18 State Baldwin effect. ☞ **[JNTU : Dec-16, Marks 2]**

Ans. : The Baldwin effect works in two steps.

1. Phenotypic plasticity allows an individual to adapt to a partially successful mutation, which might otherwise be useless to the individual. If this mutation increases inclusive fitness, it will tend to proliferate in the population. However, phenotypic plasticity is typically costly for an individual. For example, learning requires energy and time, and it sometimes involves dangerous mistakes.

2. Given sufficient time, evolution may find a rigid mechanism that can replace the plastic mechanism. Thus a behavior that was once

learned (the first step) may eventually become instinctive (the second step).

Q.19 What is Lamarckian evolution ?

Ans. :

- According to Lamarckism, the offspring of those giraffes that did succeed in transmitting an acquired extension of their necks to the next generation could obtain more food than other members of their cohort. They would thus be more numerous, which, in turn, would result in an increase of the average neck length in successive generations.
- The currently accepted view is that the genetic makeup of an individual is, in fact, unaffected by the lifetime experience of one's biological parents.

4.4 : Parallelizing Genetic Algorithms

Q.20 Explain parallelizing genetic algorithm.

Ans. :

- Genetic algorithms are naturally suited to parallel implementation. Classification relies on the computation/communication ratio.
- Coarse grain approaches to parallelization subdivide the population into somewhat distinct groups of individuals, called demes.
- Each deme is assigned to a different computational node, and a standard GA search is performed at each node.
- Fine-grained implementations typically assign one processor per individual in the population.
- Fine-grained parallel GAs are suited for massively parallel computers and consist of one spatially-structured population.
- Selection and mating are restricted to a small neighborhood, but neighborhoods overlap permitting some interaction among all the individuals. The ideal case is to have only one individual for every processing element available.
- Fig. Q.20.1 shows a schematic of a master-slave parallel GA.
- The master stores the population, executes GA operations, and distributes individuals to the slaves.

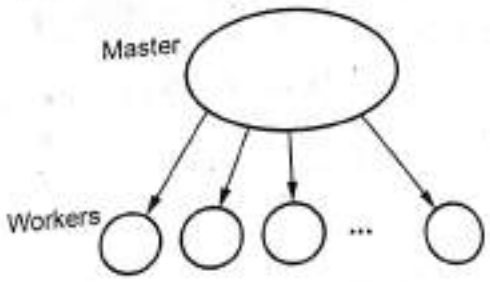


Fig. Q.20.1

The slaves only evaluate the fitness of the individuals.

4.5 : Introduction of Learning Sets of Rules

Q.21 How to learn the target function represented as a set of if-then rules?

Ans. :

- One way to learn sets of rules is to first learn a decision tree, then translate the tree into an equivalent set of rules-one rule for each leaf node in the tree.
- Second method is to use a genetic algorithm that encodes each rule set as a bit string and uses genetic search operators to explore this hypothesis space.

Q.22 What do you mean learning rule ?

Ans. :

- One of the most expressive and human readable representations for learned hypotheses is sets of production rules (if-then rules).
- Rules can be derived from other representations (e.g., decision trees) or they can be learned directly. Here, we are concentrating on the direct method.
- An important aspect of direct rule-learning algorithms is that they can learn sets of first-order rules which have much more representational power than the propositional rules that can be derived from decision trees.
- Learning first-order rules can also be seen as automatically inferring PROLOG programs from examples.

Q.23 What is sequential covering algorithms ?

Ans. :

- A covering algorithm, in the context of propositional learning systems, is an algorithm that develops a cover for the set of positive examples, that is, a set of hypotheses that account for all the positive examples but none of the negative examples.
- This is called sequential covering because it learn one rule at a time and repeat this process to gradually cover the full set of positive examples.
- The algorithm - given a set of examples:
  1. Start with an empty Cover
  2. Using Learn-One-Rule to find the best hypothesis.
  3. If the Just-Learnt-Rule satisfies the threshold then
     - Put Just-Learnt-Rule to the Cover.
     - Remove examples covered by Just-Learnt-Rule.
     - Go to step 2.
  4. Sort the Cover according to its performance over examples.
  5. Return: Cover.

Q.24 Define First-Order learning problems.

Ans. : In First-Order learning problems, the hypotheses that must be represented involve relational assertions that can be conveniently expressed using first-order representations such as horn clauses.

Q.25 What is difference between Sequential covering and Simultaneous covering ?

Ans. : Sequential covering :

- Learn just one rule at a time, remove the covered examples and repeat the process on the remaining examples.
- Many search steps, making independent decisions to select each precondition for each rule.

Simultaneous covering :

- ID3 learns the entire set of disjunct rules simultaneously as part of a single search for a decision tree.
- Fewer search steps, because each choice influences the preconditions of all rules

Q.26 Explain the algorithm of LEARN-ONE-RULE.

Ans. :

LEARN-ONE-RULE(Target _attribute, Attributes, Examples, k)

// Returns a single rule that covers some of the Examples. Conducts a general to specific greedy beam search for the best rule, guided by the PERFORMANCE metric.

- Initialize Best _hypothesis to the most general hyothesis ϕ
- Initialize Candidate _hypotheses to the set {Best _hypothesis }
- While Candidate _hypotheses is not empty, Do
  1. Generate the next more specific candidate_hypotheses

     New _Candidate _hypotheses ← new generated and specialized candidates
  2. Update Best_hypotheses

     Best _hypothesis ← h with best PERFORMANCE
  3. Update Candidate_hypotheses

  Candidate _hypotheses ← the k best members of New _Candidate _hypotheses
- Return a rule of the form

  "IF Best _hypothesis THEN prediction"

  where prediction is the most frequent value of Target _attribute among those

  Examples that match Best _hypothesis.

4.6 : Learning First-Order Rules

Q.27 Why Learn First Order rules ?

Ans. :

- Propositional logic allows the expression of individual propositions and their truth-functional combination.
- For example : propositions like Tom is a man or All men are mortal may be represented by single proposition letters such as P or Q.
- Truth functional combinations are built up using connectives, such as $\wedge$, $\vee$, -, – (for example : P$\wedge$Q)

- Inference rules are defined over propositional forms (For example: P → Q)
- Note that if P is Tom is a man and Q is All men are mortal, then the inference that Tom is mortal does not follow in propositional logic.
- First order logic allows the expression of propositions and their truth functional combination, but it also allows us to represent propositions as assertions of predicates about individuals or sets of individuals.
- For example : propositions like Tom is a man or All men are mortal may be represented by predicate-argument representations such as man(tom) or ∀ x(man(x) → mortal(x))
- Inference rules permit conclusions to be drawn about sets/individuals - e.g. mortal(tom)
- First order logic is much more expressive than propositional logic - i.e. it allows a finer-grain of specification and reasoning when representing knowledge.
- In the context of machine learning, consider learning the relational concept daughter(x, y) defined over pairs of persons x, y, where - persons are represented by attributes: <Name, Mother, Father, Male, Female >
- Training examples then have the form: <person1, person2, target_attribute_value >
- Example: << Name1 = Ann, Mother1 = Sue, Father1 = Bob, Male1 = F, Female1 = T > <Name2 = Bob, Mother2 = Gill, Father2 = Joe, Male2 = T, Female2 = F, Daughter1,2 = T >
- From such examples, a propositional rule learner such as ID3 or CN2 can only learn rules like:
- IF (Father1 = Bob) ∧ (Name2 = Bob) ∧ (Female1 = T)
- THEN Daughter1,2 = T
- This will not be useful in classifying future pairs of persons.

Q.28 Discuss about FOIL.

Ans. :

- FOIL is the natural extension of SEQUENTIAL-COVERING and LEARN-ONE-RULE to first order rule learning.
- FOIL learns first order rules which are similar to Horn clauses with two exceptions :
  1. Literals may not contain function symbols (reduces complexity of hypothesis space)
  2. Literals in body of clause may be negated (hence, more expressive than Horn clauses)
- Like SEQUENTIAL-COVERING, FOIL learns one rule at time and removes positive examples covered by the learned rule before attempting to learn a further rule.
- Unlike SEQUENTIAL-COVERING and LEARN-ONE-RULE, FOIL
  1. Only tries to learn rules that predict when the target literal is true, propositional version sought rules that predicted both true and false values of target attribute
  2. Performs a simple hill-climbing search (beam search of width one)
- FOIL searches its hypothesis space via two nested loops :
  1. The **outer loop** at each iteration adds a new rule to an overall disjunctive hypothesis. This loop may be viewed as a specific-to-general search, starting with the empty disjunctive hypothesis which covers no positive instances and stopping when the hypothesis is general enough to cover all positive examples.

2. The **inner loop** works out the detail of each specific rule, adding conjunctive constraints to the rule precondition on each iteration. This loop may be viewed as a general-to-specific search, starting with the most general precondition (empty) and stopping when the hypothesis is specific enough to exclude all negative examples.

Algorithm :

FOIL (Target_predicate, Predicates, Examples)

- Pos ← positive Examples
- Neg ← negative Examples
- Learned_rules ← { }
- while Pos, do

  Learn a NewRule

  NewRule ← most general rule possible (no preconditions)

  NewRuleNeg ← Neg

  while NewRuleNeg, do
  Add a new literal to specialize NewRule

  1. Candidate_literals ← generate candidates based on Predicates
  2. Best_literal ←

     $$\text{argmax}_{L \in} \text{Candidate\_literals Foil\_Gain(L.NewRule)}$$

  3. Add Best_literal to NewRule preconditions
  4. NewRuleNeg ← subset of NewRuleNeg that satisfies NewRule preconditions

  – Learned_rules ← Learned_rules + NewRule

  – Pos ← Pos – {members of Pos covered by NewRule}

- Return Learned_rules

Q.29 Discuss limitation of FOIL.

Ans. :

1. Search space of literals can become intractable.
2. Requires large extensional background definitions.
3. Hill-climbing search gets stuck at local optima and may not even find a consistent clause.
4. Requires complete examples to learn recursive definitions.
5. Requires a large set of closed-world negatives
6. Inability to handle logical function
7. Background predicates must be sufficient to construct definition, e.g. cannot learn reverse unless given append

4.7 : Induction as Inverted Deduction, Inverting Resolution

Q.30 Discuss briefly induction as inverted deduction.

Ans. :

- Induction is the inverse of deduction
- Given some data D and some partial background knowledge B, learning can be described as generating a hypothesis h that, together with B, explains D.
- If the training data is a set of examples of the form $<x_i, f(x_i)>$ where x_i denotes the i^{th} training example and $f(x_i)$ denotes its target value.
- Then learning is the problem of discovering h such that
  $(\forall < x_i , f(x_i) > \in D) (B \wedge h \wedge x_i)$ entails $f(x_i)$
- Example: Target concept is Child(u, v)
- Single positive example Child(Bob, Sharon) where instance is described by Male(Bob), Female(Sharon), and Father(Sharon, Bob)
- General background knowledge of Parent(u,v) Father(u,v)
- Two of the many hypothesis that satisfy $(B \wedge h \wedge x_i)$ entails $f(x_i)$ are :

  h1: Child(u,v) ← Father(v,u)

  h2: Child(u,v) ← Parent(v,u)
- Induction is, in fact, the inverse operation of deduction, and cannot be conceived to exist without the corresponding operation, so that the question of relative importance cannot arise

Q.31 What is inverting resolution ? Explain.

Ans. :

- The resolution rule is a sound and complete rule for deductive inference in first-order logic.
- Let L be an arbitrary propositional literal, and let P and R be arbitrary propositional clauses.
- The resolution rule is

$$\frac{\begin{array}{ccc} P & V & L \\ \neg L & v & R \end{array}}{\begin{array}{ccc} P & V & R \end{array}}$$

- Given the two clauses the line, conclude the clause below the line. Intuitively, the resolution rule is quite sensible.
- Given the two assertions P v L and ¬ L v R, it is obvious that either L or ¬ L must be false. Therefore, either P or R must be true. Thus, the conclusion P v R of the resolution rule is intuitively satisfying.
- This operator used in Cigol
- $C = A \vee B$ and $C_2 = B \vee D$
- Any literal present in C but not in C_1 must be present in C_2
- $C_2 = A \vee \neg D$ or $C_2 = A \vee \neg D \vee B$
- The literal that occurs in C_1 but not in C must be the literal removed by the resolution rule and therefore its negation must occur in C_2.
- Cigol uses inverse resolution with sequential covering butwith first order representations.

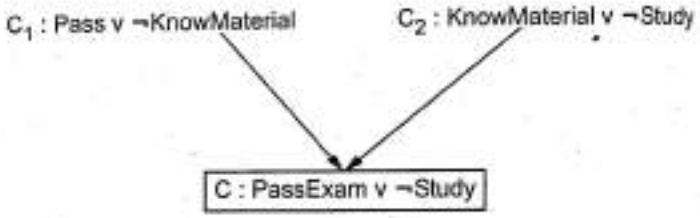


Fig. Q.31.1 Resolution

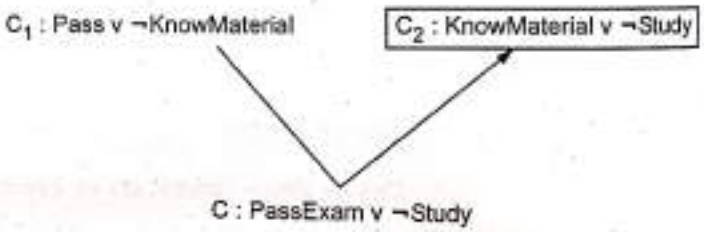


Fig. Q.31.2 Inverse resolution

4.8 : Reinforcement Learning and Q-Learning

Q.32 What is reinforcement learning ?

Ans. : • User will get immediate feedback in supervised learning and no feedback from unsupervised learning. But in the reinforced learning, you will get delayed scalar feedback.

- Reinforcement learning is learning what to do and how to map situations to actions. The learner is not told which actions to take. Fig. Q.32.1 shows concept of reinforced learning.
- Reinforced learning is deals with agents that must sense and act upon their environment. It combines

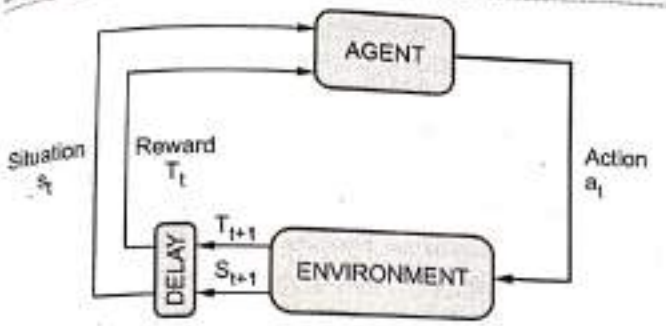


Fig. Q.32.1 Reinforced learning

classical Artificial Intelligence and machine learning techniques.

- It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal.
- Two most important distinguishing features of reinforcement learning is trial-and-error and delayed reward.
- With reinforcement learning algorithms an agent can improve its performance by using the feedback it gets from the environment. This environmental feedback is called the reward signal.
- Based on accumulated experience, the agent needs to learn which action to take in a given situation in order to obtain a desired long term goal. Essentially actions that lead to long term rewards need to reinforced. Reinforcement learning has connections with control theory, Markov decision processes and game theory.
- **Example of Reinforcement Learning :** A mobile robot decides whether it should enter a new room in search of more trash to collect or start trying to find its way back to its battery recharging station. It makes its decision based on how quickly and easily it has been able to find the recharger in the past.

Q.33 Explain elements of reinforcement learning ?

Ans. : • Reinforcement learning elements are as follows :

1. Policy
2. Reward Function
3. Value Function
4. Model of the environment

- Fig. Q.33.1 shows

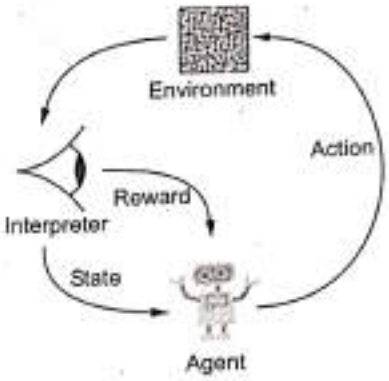


Fig. Q.33.1 : Elements of reinforcement learning

- **Policy :** Policy defines the learning agent behavior for given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.
- **Reward Function :** Reward function is used to define a goal in a reinforcement learning problem. It also maps each perceived state of the environment to a single number.
- **Value function :** Value functions specify what is good in the long run. The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.
- **Model of the environment :** Models are used for planning
- Credit assignment problem : Reinforcement learning algorithms learn to generate an internal value for the intermediate states as to how good they are in leading to the goal.
- The learning decision maker is called the agent. The agent interacts with the environment that includes everything outside the agent.
- The agent has sensors to decide on its state in the environment and takes an action that modifies its state.
- The reinforcement learning problem model is an agent continuously interacting with an environment. The agent and the environment interact in a sequence of time steps. At each time step t, the agent receives the state of the environment and a

scalar numerical reward for the previous action, and then the agent then selects an action.

- Reinforcement Learning is a technique for solving Markov Decision Problems.
- Reinforcement learning uses a formal framework defining the interaction between a learning agent and its environment in terms of states, actions, and rewards. This framework is intended to be a simple way of representing essential features of the artificial intelligence problem.

Q.34 Explain learning task.

Ans. : • Here we define one quite general formulation of the problem, based on Markov decision processes.

- In a Markov decision process (MDP) the agent can perceive a set S of distinct states of its environment and has a set A of actions that it can perform.
- A Markov decision process is a tuple $(S, A, \{P_{sa}\}, \gamma, R)$.

where :

- S is a set of actions. (For example, in autonomous helicopter flight. S might be the set of all possible positions and orientations of the helicopter.)
- A is set of actions. (For example, the set of all possible directions in which you can push the helicopter's control sticks).
- P_{sa} are the state transition probabilities. For each state $s \in S$ and action $a \in A$, P_{sa} is a distribution over the state space. We'll say more about this layer, but briefly, P_{sa} gives the distribution over what states we will transition to if we take action a in state s.
- $\gamma \in (0,1)$ is called the **discount factor.**
- $R : S \times A \rightarrow \Re$ is the **reward function.** (Rewards are sometimes also written as a function of a state S only, in which case we would have $R : S \rightarrow \Re$.

Q.35 Define Q-learning.

Ans. : Q-learning is a form of model-free reinforcement learning. It can also be viewed as a method of asynchronous dynamic programming (DP). It provides agents with the capability of learning to act optimally in Markovian domains by experiencing the consequences of actions, without requiring them to build maps of the domains

Q.36 List the advantages and disadvantage of Q-learning.

Ans. : Advantage : Converges to an optimal policy in both deterministic and nondeterministic Markov decision process

Disadvantage : Only practical on a small number of problems.

Q.37 What is Q-Learning ? Explain

Ans. : • Q-learning is a reinforcement learning technique used in machine learning. The goal of Q-Learning is to learn a policy, which tells an agent which action to take under which circumstances.

- In Q-learning , an agent tries to learn the optimal policy from its history of interaction with the environment.
- It has the ability to compute the utility of the actions without a model for the environment. It takes the help of action-value pair and the expected reward from the current action.
- During this process the agent learns to move around the environment and understand the current state which is the optimal policy by taking the action with the highest reward. Let us look at an example of this technique.
- The value Q(s,a) is defined to be the expected discounted sum of future payoffs obtained by taking action a from state s and following an optimal policy thereafter.
- Once these values have been learned, the optimal action from any state is the one with the highest Q-value.
- Consider a computational agent moving around some discrete, finite world, choosing one from a finite collection of actions at every time step. The world constitutes a controlled Markov process with the agent as a controller.
- At step n, the agent is equipped to register the state $x_n (\in X)$ of the world, an can choose its action an $(\in 2)$ accordingly.
- The agent receives a probabilistic reward r_n, whose mean value depends only on the state and action

and the state of the world changes probabilistically to y_n according to the law :

$$\text{Prob}[y_n = y | x_n, a_n] = P_{x_n y}[a_n]$$

- The task facing the agent is that of determining an optimal policy, one that maximizes total discounted expected reward.
- By discounted reward, we mean that rewards received s steps hence are worth less than rewards received now, by a factor of $y^s (0 < y < 1)$.
- The state values of x :

$$Q^{\pi}(x, a) = R_x(a) + \gamma \sum_y P_{xy}[\pi(x)] V^{\pi}(y).$$

4.9 : Non-deterministic Rewards and Actions

Q.38 Discuss about non-deterministic rewards and actions of Q-learning.

Ans. :

- In the nondeterministic case reward function r(s,a) and action transition function δ (s,a) have probabilistic outcomes.
- For example, in robot problems with noisy sensors and effectors it is often appropriate to model actions and rewards as nondeterministic.
- In such cases, the functions δ (s, a) and r(s, a) can be viewed as first producing a probability distribution over outcomes based on s and a, and then drawing an outcome at random according to this distribution.
- When these probability distributions depend solely on s and a, then we call the system a nondeterministic Markov decision process.
- In the nondeterministic case, we must first restate the objective of the learner to take into account the fact that outcomes of actions are no longer deterministic.
- The obvious generalization is to redefine the value of a policy n to be the expected value V^{π} of the discounted cumulative reward received by applying this policy

$$V^{\pi}(S_t) = E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right]$$

- The optimal policy n^* to be the policy n that maximizes $V^n(s)$ for all states s. Next we generalize definition of Q, again by taking its expected value

$$\begin{aligned} Q(s, a) &= E[r(s, a) + \gamma V^*(\delta(s, a))] \\ &= E[r(s, a)] + \gamma E[V^*(\delta(s, a))] \\ &= E[r(s, a)] + \gamma \sum_{s'} P(s'|s, a) V^*(s') \end{aligned}$$

Where P(s'|s, a) is the probability that taking action a in state s will produce the next state s'. Note we have used P(s'|s, a) here to rewrite the expected value of $V^*(\delta(s, a))$ in terms of the probabilities associated with the possible outcomes of the probabilistic δ.

4.10 : Temporal Difference Learning

Q.39 What is Temporal Difference Learning ? Explain in detail.

Ans. : • Temporal-difference (TD) learning is a combination of Monte Carlo ideas and dynamic programming (DP) ideas.

- Like Monte Carlo methods, TD methods can learn directly from raw experience without a model of the environment's dynamics.
- Like DP, TD methods update estimates based in part on other learned estimates, without waiting for a final outcome.
- The relationship between TD, DP, and Monte Carlo methods is a recurring theme in the theory of reinforcement learning.
- TD learning, which is a model-free learning algorithm, has two important properties:
  1. It doesn't require the model dynamics to be known in advance
  2. It can be applied for non-episodic tasks as well
- The algorithm takes the benefits of both the Monte Carlo method and dynamic programming (DP) into account :
  1. Like the Monte Carlo method, it doesn't require model dynamics, and
  2. Like dynamic programming, it doesn't need to wait until the end of the episode to make an estimate of the value function.

- We try to predict the state values in temporal difference learning, TD learning does not need model of the environment unlike DP.
- TD learning using something called a TD update rule for updating the value of a state :

$$V(S_t) \leftarrow V(S_t) + \alpha\,[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

- value of a previous state = value of previous state + learning_rate*(reward + discount_factor (value of current state) — value ofprevious state)
- This is the difference between the **actual reward (r + Gamma * V(s'))** and the expected reward **V(s)** multiplied by the learning rate alpha.
- The learning rate, also called step size, is useful for convergence.

Fill in the Blanks for Mid Term Exam

Q.1 The chromosome are divided into several parts called genes .

Q.2 Genetic algorithms are inspired by Darwin's theory about evolution.

Q.3 The mutation depends on the encoding as well as the Crossover

Q.4 A genetic algorithm is a search technique used in computing to find true or approximate solutions to optimization and search problems.

Q.5 Crossover is the process of taking two parent solutions and producing from them a child.

Q.6 In two-point crossover, offspring are created by substituting intermediate segments of one parent into the middle of the second parent string.

Q.7 The general-to-specific search suggested above for the LEARN-ONE-RULE algorithm is a greedy depth-first search with no backtracking.

Q.8 Q-learning is a _____ techniques used in machine learning. Reinforcement learning.

Q.9 Reward function is used to define a goal in a reinforcement learning problem.

Multiple Choice Questions for Mid Term Exam

Q.1 Genetic Algorithm are a part of _____.

- [a] evolutionary computing
- [b] inspired by Darwin's theory about evolution - "survival of the fittest"
- [c] are adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and genetics
- [d] All of the above

Q.2 Which GA operation is computationally most expensive ?

- [a] Initial population creation.
- [b] Selection of sub?population for mating.
- [c] Reproduction to produce next generation.
- [d] Convergence testing.

Q.3 Which of the following is not true for Genetic algorithms ?

- [a] It is a probabilistic search algorithm.
- [b] It is guaranteed to give global optimum solutions.
- [c] If an optimization problem has more than one solution, then it will return all the solutions.
- [d] It is an iterative process suitable for parallel programming.

Q.4 The purpose of the fitness evaluation operation is _____.

- [a] to check whether all individual satisfies the constraints given in the problem.
- [b] to decide the termination point.
- [c] to select the best individuals.
- [d] to identify the individual with worst cost function.

Q.5 Roulette wheel selection scheme is preferable when _____.

- [a] fitness values are uniformly distributed.

UNIT - V

5 Analytical Learning

5.1 : Introduction to Analytical Learning

Q.1 What is analytical learning ? [JNTU : Dec-17, Marks 2]

Ans. : In analytical learning, the input to the learner includes the same hypothesis space H and training examples D as for inductive learning. In addition, the learner is provided an additional input: A **domain theory B** consisting of background knowledge that can be used to explain observed training examples. The desired output of ,the learner is a hypothesis h from H that is consistent with both the training examples D and the domain theory B.

Q.2 What is inductive learning ?

Ans. : In inductive learning, the learner is given a hypothesis space H from which it must select an output hypothesis, and a set of training examples $D = \{(x_1, f(x_1)). ., . (x_n, f(x_n))\}$ where $f(x_i)$ is the target value for the instance x_i. The desired output of the learner is a hypothesis h from H that is consistent with these training examples.

Q.3 What is the difference between inductive and analytical learning methods ?

Ans. :

Parameters	Inductive learning	Analytical learning
Goal	Hypothesis fits data	Hypothesis fits domain theory
Justification	Statistical inference	Deductive inference
Merit	Requires little prior knowledge	Learns from scarce data
"Demerit	• Scarce data, • Incorrect bias	Imperfect domain theory

Q.4 What is domain theory ?

Ans. : A domain theory is said to be correct if each of its assertions is a truthful statement about the world. A domain theory is said to be complete with respect to a given target concept and instance space, if the domain theory covers every positive example in the instance space.

5.2 : Learning with Perfect Domain Theories : PROLOG-EBG

Q.5 What is prolog ?

Ans. : • Prolog is a logic programming language associated with artificial intelligence and computational linguistics.

- Prolog has its roots in first-order logic, a formal logic, and unlike many other programming languages, Prolog is intended primarily as a declarative programming language: the program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a query over these relations.

Q.6 What are the main properties of PROLOG-EBG algorithm ? Is it deductive or inductive ? Justify your answer.

☞ [JNTU : Dec-17, Marks-5]

Ans. :
- PROLOG-EBG is a sequential covering algorithm.
- PROLOG-EBG computes the most general rule that can be justified by the explanation, by computing the weakest pre-image of the explanation.
- PROLOG-EBG constructs intermediate features after analyzing examples.
- It is deductive learning system, which assume that domain knowledge is correct and complete.
- PROLOG-EBG produces justified general hypotheses by using prior knowledge to analyze individual examples
- PROLOG-EBG implicitly assumes that the domain theory is correct and complete. If the domain theory is incorrect or incomplete, the resulting learned concept may also be incorrect.
- The generality of the learned Horn clauses will depend on the formulation of the domain theory and on the sequence in which training examples are considered.
- In its pure form, PROLOG-EBG is a deductive, rather than inductive, learning process. That is, by calculating the weakest pre-image of the explanation it produces a hypothesis h that follows deductively from the domain theory B, while covering the training data D.

Q.7 Discuss Prolog-EBG algorithm.

☞ [JNTU : Dec-16, Marks 5]

Ans. : **PROWG-EBG(TargetConcept, TrainingExamples, DomainTheory)**

LearnedRules ← { }

Pos ← the positive examples from **TrainingExamples**

for each **PositiveExample** in Pos that is not covered by **LearnedRules**, do

1. **Explain : Explanation** ← an explanation (proof) in terms of the **DomainTheory** that **PositiveExample** satisfies the **TargetConcept**
2. **Analyze : SufficientConditions** ← the most general set of features of **PositiveExample** sufficient to satisfy the TargetConcept according to the Explanation.
3. **Rejine : LearnedRules** ← **LearnedRules + NewHornClause**, where **NewHornClause** is of the form

   TargetConcept ← **SufficientConditions**

Return **LearnedRules**

5.3 : Remarks on Explanation Based Learning

Q.8 What is explanation-based learning ?

Ans. :

- Explanation-based learning is a form of analytical learning in which the learner processes each novel training example by explaining the observed target value for this example in terms of the domain theory, analyzing this explanation to determine the general conditions under which the explanation holds refining its hypothesis to incorporate these general conditions.
- An Explanation-based Learning (EBL) system accepts an example (i.e. a training example) and explains what it learns from the example.
- The EBL system takes only the relevant aspects of the training. This explanation is translated into particular form that a problem solving program can understand. The explanation is generalized so that it can be used to solve other problems.
- The EBL module uses the results from the problem-solving trace (ie. Steps in solving problems) that were generated by the central problem solver.
- It constructs explanations using an axiomatized theory that describes both the domain and the architecture of the problem solver.
- The results are then translated as control rules and added to the knowledge base. The control knowledge that contains control rules is used to guide the search process effectively.

Q.9 List and explain Explanation-based Learning phases.

Ans. : EBL phases are as follows :

1. Problem solving

- From the example of the concept and the domain theory a solution that explains the concept is obtained
- From this resolution we are interested on all the actions performed
- These action will be the trace of the resolution, and will be used during the generalization process

2. Resolution trace analysis and filtering

- The domain determines the operational criteria that tells which are the primitive actions for the problem
- The relevance criteria will also be defined , this will allow to decide what parts of the resolution are important
- Using these two criteria the parts that need to be generalized from the resolution trace will be determined
- The filtered resolution trace will be the explanation of the example.

3. Generalization of the explanation

- The generalization of the explanation requires the substitution of constants by variables in a way that preserves the original explanation
- The usual mechanism for the generalization id the goal regression algorithm
- This algorithm consists on the variabilization of the goal and the propagation of the substitution in the explanation

4. Building the new knowledge

- The explanation has to be expressed using the primitive predicates of the domain
- The knowledge has to be translated to the representation formalism of the domain theory
- This knowledge can be new definition of the domain predicates or control rules that represent how the knowledge has to be used to solve new problems

5. Incorporating the new knowledge

- Sometimes is not enough to add the knowledge to the domain theory
- If we do not want the theory to degrade, the new knowledge has to be transformed so it can be used efficiently
- Their use can also be evaluated so it can be eliminated if it is not used frequently enough

Q.10 Explain elements of Explanation-based Learning.

Ans. : EBL elements are as follows :

1. Domain theory: Information about the specific domain of the problem
2. Goal concept: Concept we want to obtain an operational definition

3. Example : Positive example of the concept we want to learn
4. New domain Theory : The initial theory plus the new definition learned for the goal concept from the example

Q.11 Explanation determines feature relevance." Substantiate this statement with respect to explanation based learning. [JNTU : Dec-16, Marks 5]

Ans. :
- Choosing good features to represent objects can be crucial to the success of supervised machine learning algorithms
- Explanation-based learning (EBL) is a method of dynamically incorporating prior domain knowledge into the learning process by explaining training examples.
- In classical EBL, an "explanation" is a logical proof that shows how the class label of a particular labeled example can be derived from the observed inputs
- Unlike inductive methods, PROLOG-EBG produces justified general hypotheses by using prior knowledge to analyze individual examples.
- The explanation of how the example satisfies the target concept determines which example attributes are relevant, those mentioned by the explanation.
- The further analysis of the explanation, regressing the target concept to determine its weakest pre-image with respect to the explanation, allows deriving more general constraints on the values of the relevant features

Q.12 What is knowledge level learning ? Explain.

Ans. :
- The knowledge level is a level of description for computer systems
- Example of knowledge-level analytical learning is provided by considering a type of assertions known as determinations.
- Determinations assert that some attribute of the instance is fully determined by certain other attributes, without specifying the exact nature of the dependence.
- For example, consider learning the target concept "people who speak Hindi," and imagine we are given as a domain theory the single determination assertion "the language spoken by a person is determined by their nationality."
- Taken alone, this domain theory does not enable us to classify any instances as positive or negative.
- However, if we observe that "Jon, a 23- year-old left-handed US, speaks Hindi," then we can conclude from this positive example and the domain theory that "all US speak Hindi."

5.4 : Using Prior Knowledge to Alter the Search Objective

Q.13 What is prior knowledge ?

Ans. : Prior knowledge refers to all information about the problem available in addition to the training data.

Q.14 Describe TANGENT PROP algorithm ?

Ans. :
- Tangent Propagation is the name of a learning technique of an artificial neural network (ANN) which enforces soft constraints on first order partial derivatives of the output vector.
- It accommodates domain knowledge expressed as derivatives of the target function with respect to transformations of its inputs.

- The TANGENTPROP algorithm assumes various training derivatives of the target function are also provided.
- For example, if each instance xi is described by a single real value, then each training example may be of the form $\left(x_i, f(x_i), \left.\frac{\partial f(x)}{\partial x}\right|_{x_i} \right)$.
- Here $\left.\frac{\partial f(x)}{\partial x}\right|_{x_i}$ denotes the derivative of the target function f with respect to x evaluated at the point $x = x_i$.
- To develop an intuition for the benefits of providing training derivatives as well as training values during learning, consider the simple learning task depicted in Fig. Q.14.1.

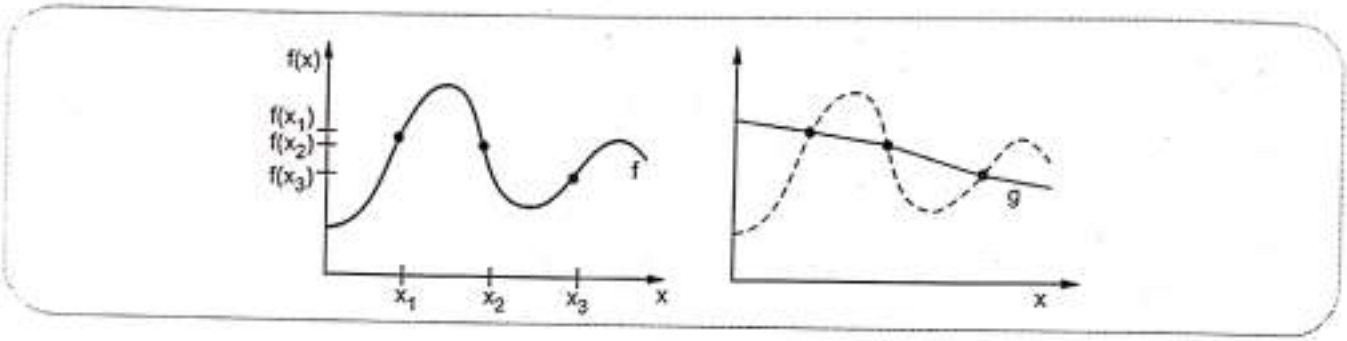


Fig. Q.14.1.

- The task is to learn the target function f shown in the leftmost plot of the figure, based on the three training examples shown: $(x_1, f(x_1))$, $(x_2, f(x_2))$, and $(x_3, f(x_3))$.
- Given these three training examples, the BACKPROPAGATATION algorithm can be expected to hypothesize a smooth function, such as the function g depicted in the middle plot of the figure.
- In TANGENTPROP an additional term is added to the error function to penalize discrepancies between the trainin4 derivatives and the actual derivatives of the learned neural network function f.

5.5 : Using Prior Knowledge to Augment Search Operators

Q.15 What is difference between first-order inductive learner (FOIL) and First Order Combined Learner (FOCL) ?

Ans. : FOIL generates each candidate specialization by adding a single new literal to the clause preconditions. FOCL uses this same method for producing candidate specializations, but also generates additional specializations based on the domain theory.

Q.16 What is FOCL ? Explain in detail.

Ans. :

- FOCL uses the domain theory to increase the number of candidate specializations considered at each step of the search for a single Horn clause.
- FOCL expands its current hypothesis h using the following two operators:

1. For each operational literal that is not part of h, create a specialization of h by adding this single literal to the preconditions.

2. Create an operational, logically sufficient condition for the target concept according to the domain theory. Add this set of literals to the current preconditions of h.

- FOCL first selects one of the domain theory clauses whose head (postcondition) matches the target concept.
- If there are several such clauses, it selects the clause whose body (preconditions) have the highest information gain relative to the training examples of the target concept.
- FOCL learns Horn clauses of the form (c O_i ^ O_b^ O_f)

  where c is the target concept,

  O_i is an initial conjunction of operational literals added one at a time by the first syntactic operator,

  O_b is a conjunction of operational literals added in a single step based on the domain theory, and

  O_f is a final conjunction of operational literals added one at a time by the first syntactic operator.

- Any of these three sets of literals may be empty.
- FOCL uses both a syntactic generation of candidate specializations and a domain theory driven generation of candidate specializations at each step in the search.
- The algorithm chooses among these candidates based solely on their empirical support over the training data.
- Thus, the domain theory is used in a fashion that biases the learner, but leaves final search choices to be made based on performance over the training data.

5.6 : Combining Inductive and Analytical Learning

Q.17 Write specific properties of a learning method.

Ans. :

Properties include:

a. Given no domain theory, it should learn at least as effectively as purely inductive methods.

b. Given a perfect domain theory, it should learn at least as effectively as purely analytical methods.

c. Given an imperfect domain theory and imperfect training data, it should combine the two to outperform either purely inductive or purely analytical methods.

d. It should accommodate an unknown level of error in the training data.

e. It should accommodate an unknown level of error in the domain theory.

5.7 : Using Prior Knowledge to Initialize the Hypothesis

Q.18 Write KBANN algorithm to explain usage of prior knowledge to reduce complexity.

[JNTU : Dec-17, Marks 5]

Ans. :

- KBANN(Domain-Theory, Training_Examples)
- Domain-Theory: Set of propositional, nonrecursive Horn clauses.
- Training example: Set of (input output) pairs of the target function.
- Analytical step: Create an initial network equivalent to the domain theory.

1. For each instance attribute create a network input.
2. For each Horn clause in the Domain-Theory, create a network unit as follows:
   - Connect the inputs of this unit to the attributes tested by the clause antecedents.
   - For each non-negated antecedent of the clause, assign a weight of W to the corresponding sigmoid unit input.
   - For each negated antecedent of the clause, assign a weight of - W to the corresponding sigmoid unit input.
   - Set the threshold weight wo for this unit to -(n - 0 .5)W, where n is the number of non-negated antecedents of the clause.
3. Add additional connections among the network units, connecting each network unit at depth i from the input layer to all network units at depth i + 1. Assign random near-zero weights to these additional connections.

Inductive step: Refine the initial network.

4. Apply the BACKPROPAGATION algorithm to adjust the initial network weights to fit the Training-Examples.

Fill in the Blanks for Mid Term Exam

Q.1 PROLOG-EBG is a sequential covering algorithm.

Q.2 KBANN stands for Knowledged-based Artificial neural network.

Q.3 SOAR uses a variant of explanation-based learning called _____ to extract the general conditions under which the same explanation applies. chunking

Q.4 In its pure form, PROLOG-EBG is a deductive, rather than _____ process inductive learning

Q.5 PROLOG-EBG computes the weakest pre-image of the target concept with respect to the explanation, using a general procedure called regression

Q.6 domain theory is said to be correct if each of its assertions is a truthful statement about the world.

Q.7 A domain theory is said to be complete with respect to a given target concept and instance space.

Q.8 Analytical learning uses prior knowledge and deductive reasoning to augment the information provided by the training examples, so that it is not subject to these same bounds.

Q.9 In two-pot crossover, offspring are created by substituting intermediate segments of one parent into the middle of the second parent string

Q.10 The general-to-specific search suggested above for the CO-SOAR algorithm is a greedy depth-first search with no backtracking.

Q.11 First order Horn clauses may also refer to variables in the preconditions that do not occur in the postconditions

Q.12 In explanation-based learning, prior is used to analyse, or explain, how each observed training example satisfies the target concept.

Q.13 Decision tree learning, neural network learning, inductive logic programming, and genetic algorithms are all examples of inductive methods that operate in fashion.

Q.14 Explanation-based learning is a form of analytical learning in which the learner processes each novel training example

Q.15 PROLOG-EBG is an explanation-based learning algorithm that uses first-order Horn clauses to represent both its d_____ and its learned hypotheses.

Mutliple Choice Questions for Mid Term Exam

Q.1 In _____, the learner must output a hypothesis that is consistent with both the training data and the domain theory.

- a analytical learning
- b inductive learning
- c deductive learning
- d none of these

Q.2 _____ crossover combines bits sampled uniformly from the two parents.

- a single point
- b two point
- c uniform
- d all of these

Q.3 A domain theory B consisting of _____ that can be used to explain observed training examples.

- a Prior knowledge
- b background knowledge
- c training examples
- d none of these

Q.4 PROLOG-EBG is an explanation-based learning algorithm that uses _____ Horn clauses to represent both its domain theory and its learned hypotheses.

- a First order
- b second order
- c no order
- d all of these

SOLVED MODEL QUESTION PAPER

Machine Learning (As per R16 Pattern)

IVth Year B. Tech., Sem - I [CSE] Professional Elective - III
IVth Year B. Tech., Sem - II [ECE] Professional Elective - V

Time : 3 Hours] [Maximum Marks : 75

Note : This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer all questions in Part A. Part B consists of 5 Units. Answer any one full question from each unit. Each question carries 10 marks and may have a, b, c as sub questions.

PART - A (Marks 25)

Q.1 a) *Define machine learning.* **(Refer Q.2 of Chapter 1)** [2]

b) *Describe List-Then-Eliminate Algorithm.* **(Refer Q.27 of Chapter 1)** [3]

c) *List advantages of Neural Networks* **(Refer Q.4 of Chapter 2)** [2]

d) *Differentiate between sample error and true error.* **(Refer Q.33 of Chapter 2)** [3]

e) *What is maximum likelihood estimation ?* **(Refer Q.8 of Chapter 3)** [2]

f) *What are the features of Bayesian learning methods ?* **(Refer Q.2 of Chapter 3)** [3]

g) *Compare and contract genetic algorithm with traditional algorithm.* **(Refer Q.6 of Chapter 4)** [2]

h) *Explain learning task.* **(Refer Q.34 of Chapter 4)** [3]

i) *What is analytical learning ?* **(Refer Q.1 of Chapter 5)** [2]

j) *What is knowledge level learning ? Explain.* **(Refer Q.12 of Chapter 5)** [3]

PART - B (Marks 25)

Q.2 a) *What are T, P, E ? How do we formulate a machine learning problem ?* **(Refer Q.4 of Chapter 1)** [5]

b) *Explain candidate elimination algorithm with example.* **(Refer Q.30 of Chapter 1)** [5]

OR

Q.3 a) *Explain Gini Index and Entropy of dicision tree algorithm.* **(Refer Q.45 of Chapter 1)** [5]

b) *Explain ID3 algorithm.* **(Refer Q.47 of Chapter 1)** [5]

Q.4 a) *Explain gradient descent algorithm. What is steepest descent algorithm?* **(Refer Q.15 of Chapter 2)** [5]

b) *What is RNN ?* **(Refer Q.30 of Chapter 2)** [5]

OR

Q.5 a) *Explain type-I and type-II errors.* **(Refer Q.48 of Chapter 2)** [5]

b) *Explain delta learning rule for multiperceptron layer.* **(Refer Q.16 of Chapter 2)** [5]

Q.6 a) *Briefly discuss least square method. List disadvantages of least square method.* **(Refer Q.9 of Chapter 3)** [5]

b) *What is case-based reasoning ? Explain its steps.* **(Refer Q.39 of Chapter 3)** [5]

OR

Q.7 a) *What is Bayes theorem ? How to select Hypotheses ?* **(Refer Q.4 of Chapter 3)** [5]

b) *Explain VC dimension.* **(Refer Q.24 of Chapter 3)** [5]

Q.8 a) *Discuss about FOIL.* **(Refer Q.28 of Chapter 4)** [5]

b) *Explain how genetic algorithms are different from evolutionary programming.* **(Refer Q.17 of Chapter 4)** [5]

OR

Q.9 a) *What is Temporal Difference Learning ? Explain in detail.* **(Refer Q.39 of Chapter 4)** [5]

b) *Explain elements of reinforcement learning ?* **(Refer Q.33 of Chapter 4)** [5]

Q.10 a) *Write KBANN algorithm to explain usage of prior knowledge to reduce complexity.* **(Refer Q.18 of Chapter 5)** [5]

b) *Describe TANGENT PROP algorithm ?* **(Refer Q.14 of Chapter 5)** [5]

OR

Q.11 a) *List and explain Explanation-based Learning phases.* **(Refer Q.9 of Chapter 5)** [5]

b) *What is FOCL ? Explain in detail.* **(Refer Q.16 of Chapter 5)** [5]

END...

DECEMBER - 2020 [138CY] [R16]
Machine Learning

Solved Paper
B.Tech., IV - II [ECE]

Time : 2 Hours] **[Maximum Marks : 75**

Answer any Five Questions.
All Questions Carry Equal Marks.

Q.1 *Derive an example to explain the working of candidate eliminate algorithm.*
(Refer Q.28 and Q.30 of Chapter - 1) **[15]**

Q.2 *You are stranded on a deserted island. Mushrooms of various types grow widely all over the island, but no other food is anywhere to be found. Some of the mushrooms have been determined as poisonous and others as not (determined by your former companions' trial and error). You are the only one remaining on the island. You have the following data to consider :* **[15]**

Example	Note Heavy	Smelly	Spotted	Smooth	Edible
A	1	0	0	0	1
B	1	0	1	0	1
C	0	1	0	1	1
D	0	0	0	1	0
E	1	1	1	0	0
F	1	0	1	1	0
G	1	0	0	1	0
H	0	1	0	0	0
U	0	1	1	1	?
V	1	1	0	1	?
W	1	1	0	0	?

You know whether or not mushrooms A through H are poisonous, but you do not know about U through W. Classify mushrooms U, V and W. using the decision tree as poisonous or not poisonous.

Ans : Considering only the data for mushrooms A through H, what is the entropy of edible.

Entropy :

$$H_{\text{Edible}} = H(3+, 5-) \overset{\text{def}}{=} -\frac{3}{8}\cdot\log_2\frac{3}{8} - \frac{5}{8}\cdot\log_2\frac{5}{8}$$

$$= \frac{3}{8}\cdot\log_2\frac{8}{3} + \frac{5}{8}\cdot\log_2\frac{8}{5}$$

$$= \frac{3}{8}\cdot 3 - \frac{3}{8}\cdot\log_2 3 + \frac{5}{8}\cdot 3 - \frac{5}{8}\cdot\log_2 5$$

$$= 3 - \frac{3}{8}\cdot\log_2 3 - \frac{5}{8}\cdot\log_2 5 \approx 0.9544$$

- Decision tree to classify mushrooms as poisonous or not :

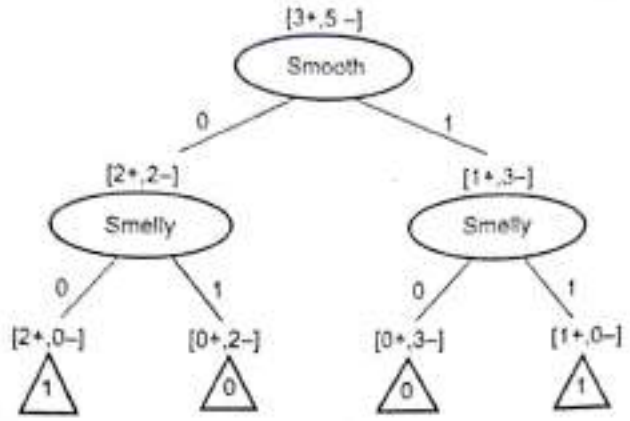


Fig. 1

- Classify mushrooms U, V and W using the decision tree as poisonous or not poisonous.

Classification of test instances :

U	Smooth = 1, Smelly = 1 ⇒ Edible = 1
V	Smooth = 1, Smelly = 1 ⇒ Edible = 1
W	Smooth = 0, Smelly = 1 ⇒ Edible = 0

Q.3 *Explain in detail perceptron based artificial neural network system, its representation and training rule.* [15]

Ans. : The perceptron is a feed-forward network with one output neuron that learns a separating hyper-plane in a pattern space. The perceptron is a classic learning algorithm for the neural model of learning. Here weights are adjusted to minimize error whenever the computed output does not match the target output.

- The perceptron learning rule is a method for finding the weights in a network. We consider the problem of supervised learning for classification although other types of problems can also be solved.
- A nice feature of the perceptron learning rule is that if there exist a set of weights that solve the problem, then the perceptron will find these weights. This is true for either binary or bipolar representations.
- The perceptron is a kind of a single - layer artificial network with only one neuron. The perceptron is a network in which the neuron unit calculates the linear combination of its real-valued or boolean inputs and passes it through a threshold activation function.
- Fig. 2 shows the basic perceptron. The perceptron is sometimes referred to a Threshold Logic Unit (TLU) since it discriminates the data depending on whether the sum is greater than the threshold value.

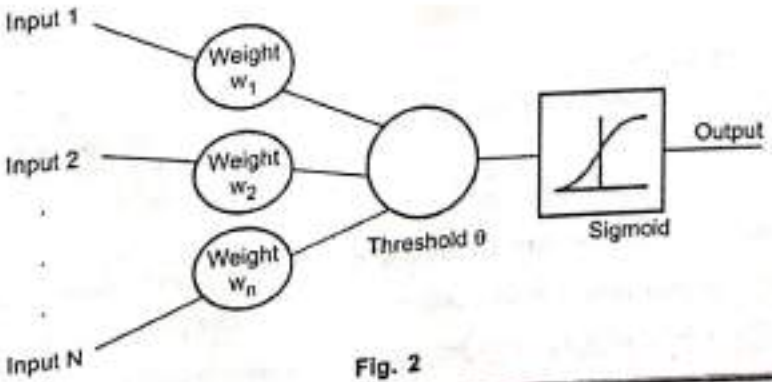


Fig. 2

- The output of the neuron is a linear combination of the inputs rescaled by the synaptic weights.
- Learning is initiated by making adjustments to the relevant connection strengths and a threshold value.
- Here we consider only two class problem. Here output layer usually has only a single node. For an n-class problem (n > 3), the output layer usually has n-nodes, each corresponding to a class and the output node with the largest value indicates which class the input vector belongs to.
- In the first stage, the linear combination of inputs is calculated. Each value of input array is associated with its weight value, which is normally between 0 and 1. Also the summation function often takes an extra input value theta with weight value of 1 to represent threshold or bias of a neuron.
- In the simplest case the network has only two inputs and a single output. The output of the neuron is :

$$y = f\left(\sum_{i=1}^{2} w_1 x_1 + b\right)$$

- Suppose that the activation function is a threshold then,

$$f = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{if } s \leq 0 \end{cases}$$

- The perceptron can represent most of the primitive boolean functions : AND, OR, NAND and NOR but cannot represent XOR.
- In single layer perceptron, initial weight values are assigned randomly because it does not have previous knowledge. It sum all the weighted inputs. If the sum is greater than the threshold value then it is activated i.e. output = 1.

Output

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n > \theta \Rightarrow 1$$

$$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n \leq \theta \Rightarrow 0$$

- The input values are presented to the perceptron, and if the predicted output is the same as the desired output, then the performance is considered satisfactory and no changes to the weights are made.
- If the ouput does not match the desired output, then the weights need to be changed to reduce the error.
- The weight adjustement is done as follows :

$$\Delta w = \eta \times d \times x$$

where x = Input data

d = Predicted output and desired output

η = Learning rate

- If the output of the perceptron is correct then we do not take any action. If the output is incorrect then the weight vector is $w \rightarrow w + \Delta w$.
- The process of weight adaptation is called **learning**.

Q.4 *Suppose hypothesis h commits r = 10 errors over a sample of n = 65 independently drawn examples.*

a) What is the variance and standard deviation for number of true error rate error D(h) ?

b) What is the 90 % confidence interval (two-sided) for the true error rate ? **[7 + 8]**

Ans. : a) Variance and standard deviation

$$p = \frac{r}{N} = \frac{10}{65} = 0.1538$$

$$\text{Variance} = np(1-p) = 65 \times 0.1538\ (1 - 0.1538) = 9.997 \times 0.8462 = 8.459$$

$$sd(h) = \sqrt{((p)\times(1-p))/n)} = \sqrt{((0.1538)\times(1-0.15381))/65)} = \sqrt{0.00}$$

$$SD(h) = 0.0447$$

b) 90% confidence interval two-sided for the true error rate :

$$\text{ERRORs}(h) = p \pm 1.64\ (sd(h)) = [0.0804531, 0.227239]$$

Q.5 *Explain Bayesian belief network and conditional independence with example.*
(Refer Q.17 and Q.18 of Chapter - 3) **[15]**

Q.6 *Describe K-nearest Neighbour learning algorithm for continues (real) valued target function.* **[15]**

Ans. : K-nearest neighbour learning algorithm for continues valued target function.

- The K-nearest neighbor (KNN) is a classical classification method and requires no training effort, critically depends on the quality of the distance measures among examples.
- The KNN classifier uses Mahalanobis distance function. A sample is classified according to the majority vote of its nearest K training samples in the feature space. Distance of a sample to its neighbors is defined using a distance function.
- All instances correspond to points in the n-D space. The nearest neighbor are defined in terms of Euclidean distance. The target function could be discrete or real valued.
- For discrete-valued, the k-NN returns the most common value among the k training examples nearest to xq.
- Voronoi diagram : the decision surface induced by 1-NN for a typical set of training examples.
- The k-NN algorithm for continuous-valued target functions. In nearest-neighbor learning the target function may be either discrete-valued or real-valued.
- Let us first consider learning discrete-valued target functions of the form $f : R^N \rightarrow V$, where V is the finite set $\{v_1, \ldots, v_s\}$.
- The k-NN algorithm for approximation a discrete-valued target function is given in Fig. 3 As shown there, the value $f(x_q)$ returned by this algorithm as its estimate of $f(x_q)$ is just the most common value of f among the k training examples nearest to x_q.
- If we choose k = 1, then the 1-NN algorithm assigns to $f(x_q)$ the value $f(x_i)$ where x_i is the training instance nearest to x_q. For larger values of k, the algorithm assigns the most common value among the k nearest training examples.

Fig. 3

- Fig. 3 shows the operation of the k-NN algorithm for the case where the instances are points in a two-dimensional space and where the target function is Boolean valued. The positive and negative training examples are shown by "+" and "–" respectively. A query point x_q is shown as well.
- Also refer Q.31 and Q.32 of Chapter - 3.

Q.7 *With an example explain the parallelizing genetic algorithms.* **(Refer Q.20 of Chapter - 4)** [15]

Q.8 *Describe in detail about inductive analytical approaches to learning.* [15]

Ans. : In analytical learning, the input to the learner includes the same hypothesis space H and training examples D as for inductive learning. In addition, the learner is provided an additional input : A domain theory B consisting of background knowledge that can be used to explain observed training examples. The desired output of, the learner is a hypothesis h from H that is consistent with both the training examples D and the domain theory B.

- In inductive learning, the learner is given a hypothesis space H from which it must select an output hypothesis, and a set of training examples $D = \{(x_1, f(x_1)) \ldots, (x_n, f(x_n))\}$ where $f(x_i)$ is the target value for the instance x_i. The desired output of the learner is a hypothesis h from H that is consistent with these training examples.
- Learning problem is defined as :
  a) A set of training examples D, possibly containing errors
  b) A domain theory B, possibly containing errors
  c) A space of candidate hypotheses H
- A hypothesis that best fits the training examples and domain theory.
- $error_D(h)$ is defined to be the proportion of examples from D that are misclassified by h. Let us define the error $error_B(h)$ of h with respect to a domain theory B to be the probability that h will disagree with B on the classification of a randomly drawn instance.
- We can attempt to characterize the desired output hypothesis in terms of these errors. For example, we could require the hypothesis that minimizes some combined measure of these errors, such as

$$\underset{h \in H}{\text{argmin}} \; k_D error_D(h) + k_B error_B(h)$$

- While this appears reasonable at first glance, it is not clear what values to assign to k_D and k_B to specify the relative importance of fitting the data versus fitting the theory. If we have a very poor theory and a great deal of reliable data, it will be best to weight $error_D(h)$ ore heavily.
- Given a strong theory and a small sample of very noisy data, the best results would be obtained by weighting $error_B(h)$ more heavily.
- Of course if the learner does not know in advance the quality of the domain theory or training data, it will be unclear how it should weight these two error components.
- An alternative perspective : Bayes theorem computes this posterior probability based on the observed data D, together with prior knowledge in the form of P(h), P(D), and P(D|h).
- Thus we can think of P(h), P(D), and P(D|h) as a form of background knowledge or domain theory, and user can think of Bayes theorem as a method for weighting this domain theory, together with the observed data D, to assign a posterior probability P(h|D) to h.
- The Bayesian view is that one should simply choose the hypothesis whose posterior probability is greatest, and that Bayes theorem provides the proper method for weighting the contribution of this prior knowledge and observed data.

- Unfortunately, Bayes theorem implicitly assumes perfect knowledge about the probability distributions P(h), P(D), and P(D|h).
- When these quantities are only imperfectly known, Bayes theorem alone does not prescribe how to combine them with the observed data.
- Hypothesis Space Search :

1. Use prior knowledge to derive an initial hypothesis from which to begin the search
2. Use prior knowledge to alter the objective of the hypothesis space search
3. Use prior knowledge to alter the available search steps.

END...