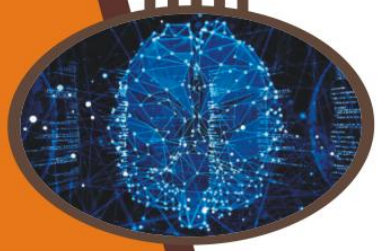
DECODE®

A Guide for Engineering Students

Neural Networks & Deep Learning

Sub. Code : CS864PE

JNTUH-R16
B.TECH., IV-II (CSE/IT)
PROFESSIONAL ELECTIVE - VI



- Written by Popular Authors of Text Books of Technical Publications
- Covers Entire Syllabus
- Question - Answer Format
- Exact Answers & Solutions
- Fill in the Blanks with Answers for Mid Term Exam
- MCQ'S with Answers for Mid Term Exam
- Short Answered Questions
- Solved Model Question Paper (R-16 Pattern)

TECHNICAL® PUBLICATIONS
SINCE 1993
An Up-Thrust for Knowledge

I. A. Dhotre

SUBJECT CODE : CS864PE

JNTUH - R16

B.Tech., IV-II (CSE / IT)

Professional Elective - VI

NEURAL NETWORKS & DEEP LEARNING

Iresh A. Dhotre

M.E. (Information Technology)

Ex-Faculty, Sinhgad College of Engineering,

Pune.

FEATURES

- ☑ Written by Popular Authors of Text Books of Technical Publications
- ☑ Covers Entire Syllabus ☑ Question - Answer Format ☑ Exact Answers & Solutions
- ☑ Important Points to Remember ☑ Fill in the Blanks with Answers for Mid Term Exam
- ☑ MCQ's with Answers for Mid Term Exam
- ☑ Short Answered Questions ☑ Solved Model Question Paper [R-16 Pattern]

DECODE®

A Guide For Engineering Students

DECODE®

A Guide For Engineering Students

NEURAL NETWORKS & DEEP LEARNING

SUBJECT CODE : CS864PE

B.Tech., IV-II [CSE / IT] Professional Elective - VI

Published by :

TECHNICAL® PUBLICATIONS SINCE 1993 An Up-Thrust for Knowledge

Amit Residency, Office No.1, 412, Shaniwar Peth,
Pune - 411030, M.S. INDIA Ph.: +91-020-24495496/97,
Email : sales@technicalpublications.org Website : www.technicalpublications.org

Printer :

Yogiraj Printers & Binders, Sr.No. 10/1A,
Ghule Industrial Estate, Nanded Village Road, Tal. - Haveli, Dist. - Pune - 411041.

First Edition : 2020

Price : ₹ 95/-

ISBN 978-93-89750-67-6

JNTUH 16

SYLLABUS

Neural Networks and Deep Learning (CS864PE)

UNIT - I

Artificial Neural Networks : Introduction, Basic models of ANN, Important terminologies, Supervised Learning Networks, Perceptron Networks, Adaptive Linear Neuron, Backpropagation Network, Associative Memory Networks, Training Algorithms for pattern association, BAM and Hopfield Networks. **(Chapter - 1)**

UNIT - II

Unsupervised Learning Network : Introduction, Fixed Weight Competitive Nets, Maxnet, Hamming Network, Kohonen Self-Organizing Feature Maps, Learning Vector Quantization, Counter Propagation Networks, Adaptive Resonance Theory Networks, Special Networks - Introduction to various networks. **(Chapter - 2)**

UNIT - III

Introduction to Deep Learning : Historical Trends in Deep learning, Deep Feed - forward networks, Gradient-Based learning, Hidden Units, Architecture Design, Back-Propagation and Other Differentiation Algorithms. **(Chapter - 3)**

UNIT - IV

Regularization for Deep Learning : Parameter norm Penalties, Norm Penalties as Constrained Optimization, Regularization and Under-Constrained Problems, Dataset Augmentation, Noise Robustness, Semi-Supervised learning, Multi-task learning, Early Stopping, Parameter Typing and Parameter Sharing, Sparse Representations, Bagging and other Ensemble Methods, Dropout, Adversarial Training, Tangent Distance, Tangent Prop and Manifold, Tangent Classifier. **(Chapter - 4)**

UNIT - V

Optimization for Train Deep Models : Challenges in Neural Network Optimization, Basic Algorithms, Parameter Initialization Strategies, Algorithms with Adaptive Learning Rates, Approximate Second-Order Methods, Optimization Strategies and Meta-Algorithms.

Applications : Large-Scale Deep Learning, Computer Vision, Speech Recognition, Natural Language Processing. **(Chapter - 5)**

TABLE OF CONTENTS

UNIT - I

1 Artificial Neural Networks

1.1 : 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 Classify different types of neurons.

☞ [JNTU : May-17, Marks 3]

Ans. : • Three major neuron groups make up this classification: multipolar, bipolar, and unipolar

- Unipolar neurons have a single short process that emerges from the cell body and divides T-like into proximal and distal branches.
- Bipolar neurons have two processes, an axon and a dendrite, that extend from opposite ends of the soma.
- Multipolar neurons, the most common type, have one axon and two or more dendrites.

Q.6 Explain useful properties and capabilities of neural network.

Ans. :

1. **Nonlinearity :** An artificial neuron can be linear or nonlinear. A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear.
2. **Adaptivity :** Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment.
3. **Contextual Information :** Knowledge is represented by the very structure and activation state of a neural network
4. **Evidential Response :** In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also about the confidence in the decision made
5. **Uniformity of Analysis and Design :** Neural networks enjoy universality as information processors
6. **VLSI Implement-ability :** The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks.

Q.7 With the help of a neat diagram, explain the analogy of a biological neuron.

☞ [JNTU : May-17, Marks 5]

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

- Fig. Q.7.1 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.
- 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.8 List the three basic elements of the neural model.

Ans. : Basic elements of the neural model are synapses or connecting links, adder and activation function.

Q.9 Define dendrites, soma, Axon and synapses.

Ans. :

- **Dendrites :** They are tree-like branches, responsible for receiving the information from other neurons it is connected to. In other sense, we can say that they are like the ears of neuron.
- **Soma :** It is the cell body of the neuron and is responsible for processing of information, they have received from dendrites.
- **Axon :** It is just like a cable through which neurons send the information.
- **Synapses :** It is the connection between the axon and other neuron dendrites.

Q.10 What are the characteristics and application of ANN ?

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.

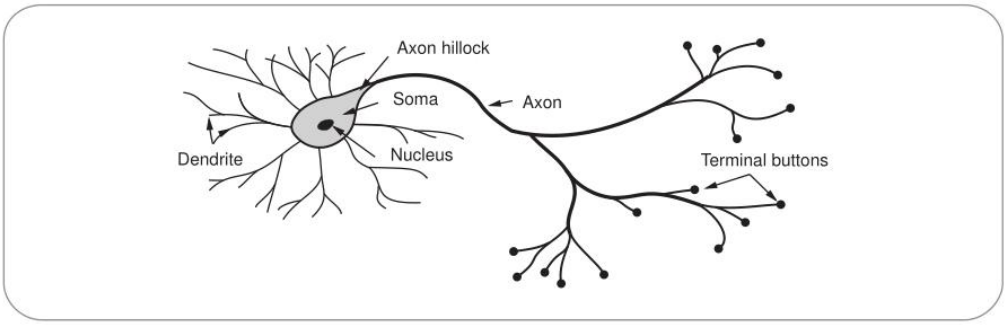


Fig. Q.7.1 Schematic of biological neuron

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.11 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.12 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.13 Explain with diagram representation of neural network.

- Fig. Q.13.1 shows the neural network representation.

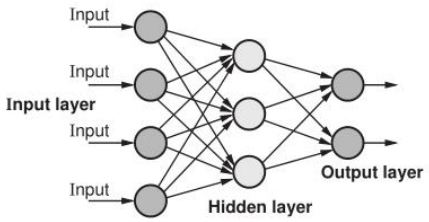


Fig. Q.13.1 Artificial neural network

- 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.14 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.

Q.15 Explain (i) Integrated and Fire Neuron model (ii) Spiking Neuron model.

☞ [JNTU : May-17, Marks 5]

Ans. : i) Integrated and Fire Neuron

- The integrate-and-fire neuron model is one of the most widely used models for analysing the behavior of neural systems.
- It describes the membrane potential of a neuron in terms of the synaptic inputs and the injected current that it receives.
- An action potential (spike) is generated when the membrane potential reaches a threshold, but the actual changes associated with the membrane voltage and conductance driving the action potential do not form part of the model.
- Fig Q.15.1 shows integrated and fire neuron model.
- The basic circuit of an integrate-and-fire model consists of a capacitor C in parallel with a resistor R driven by a current I(t).
- The driving current can be split into two components, $I(t) = I_R + I_C$.
- The first component is the resistive current I_R which passes through the linear resistor R. It can be calculated from Ohm's law as $I_R = u/R$ where u is the voltage across the resistor.
- The second component I_C charges the capacitor C. From the definition of the capacity as $C = q/u$ (where q is the charge and u the voltage), we find a capacitive current $I_C = C\ du/dt$.

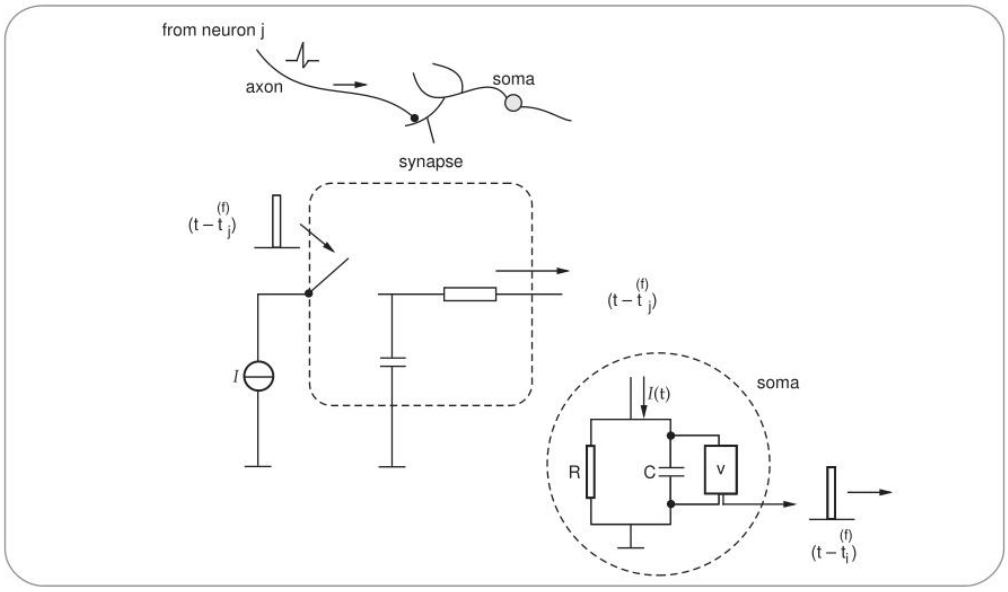


Fig. Q.15.1

- The synaptic inputs to the neuron are considered to be stochastic and are described as a temporally homogeneous Poisson process.
- Methods and results for both current synapses and conductance synapses are examined in the diffusion approximation, where the individual contributions to the postsynaptic potential are small.

ii) Spiking Neuron model

- Detailed conductance-based neuron models can reproduce electrophysiological measurements to a high degree of accuracy, but because of their intrinsic complexity these models are difficult to analyse.
- For this reason, simple phenomenological spiking neuron models are highly popular for studies of neural coding, memory, and network dynamics.
- A simple spiking neural model can carry out computations over the input spike trains under several different modes.
- Thus, spiking neurons compute when the input is encoded in temporal patterns, firing rates, firing rates and temporal correlations, and space rate codes.
- An essential feature of the spiking neurons is that they can act as coincidence detectors for the incoming pulses, by detecting if they arrive in almost the same time
- Spikes are generated whenever the membrane potential u crosses some threshold v from below. The moment of threshold crossing defines the firing time $t^{(f)}$

$$t^{(f)}: u(t^{(f)}) = V \quad \text{and} \quad \left.\frac{du(t)}{dt}\right|_{t = t^{(t)}} > 0$$

1.2 : Supervised Learning Networks

Q.16 Define supervised learning.

Ans. : Supervised learning in which the network is trained by providing it with input and matching output patterns. These input-output pairs are usually provided by an external teacher. A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier or a regression function.

Q.17 Explain supervised learning.

Ans. : Supervised learning : • Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. The task of the supervised learner is to predict the output behavior of a system for any set of input values, after an initial training phase.

- **Supervised learning** in which the network is trained by providing it with input and matching output patterns. These input-output pairs are usually provided by an external teacher.
- Human learning is based on the past experiences. A computer does not have experiences.
- A computer system learns from data, which represent some "past experiences" of an application domain.
- To learn a target function that can be used to predict the values of a discrete class attribute, e.g., approve or not-approved, and high-risk or low risk. The task is commonly called : Supervised learning, Classification or inductive learning.
- Training data includes both the input and the desired results. For some examples the correct results (targets) are known and are given in input to the model during the learning process. The construction of a proper training, validation and test set is crucial. These methods are usually fast and accurate.
- Have to be able to generalize : give the correct results when new data are given in input without knowing a priori the target.
- Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object and a desired output value.
- A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier or a regression function. Fig. Q.17.1 shows supervised learning process.
- The learned model helps the system to perform task better as compared to no learning.
- Each input vector requires a corresponding target vector.

  Training Pair = (Input Vector, Target Vector)

- Fig. Q.17.2 shows input vector.

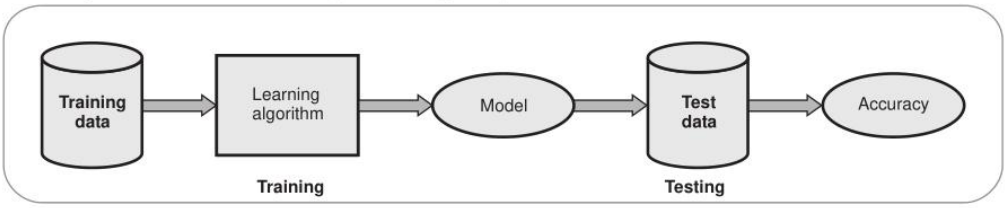


Fig. Q.17.1 Supervised learning process

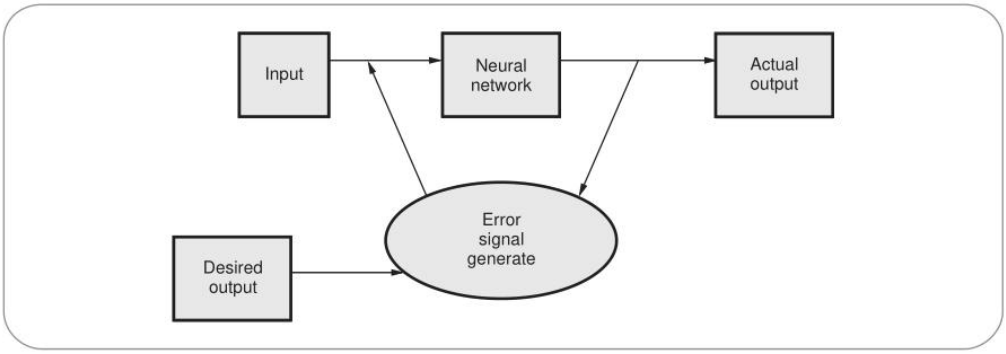


Fig. Q.17.2 Input vector

- Supervised learning denotes a method in which some input vectors are collected and presented to the network. The output computed by the net-work is observed and the deviation from the expected answer is measured. The weights are corrected according to the magnitude of the error in the way defined by the learning algorithm.
- Supervised learning is further divided into methods which use reinforcement or error correction. The perceptron learning algorithm is an example of supervised learning with reinforcement.
- In order to solve a given problem of supervised learning, following steps are performed :
  1. Find out the type of training examples.
  2. Collect a training set.
  3. Determine the input feature representation of the learned function.
  4. Determine the structure of the learned function and corresponding learning algorithm.
  5. Complete the design and then run the learning algorithm on the collected training set.
  6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

Q.18 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.19 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.

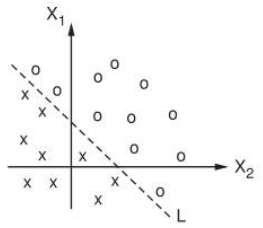


Fig. Q.19.1

- 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.20 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 one way to use a two-input perceptron to implement the AND function.

- Perceptron can represent all of the primitive Boolean functions AND, OR, NAND (1 AND), and NOR (1 OR). Unfortunately, however, some Boolean functions cannot be represented by a single perceptron, such as the XOR function whose value is 1 if and only if $x_1 \# x_2$.
- The decision surface represented by a two-input perceptron.

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

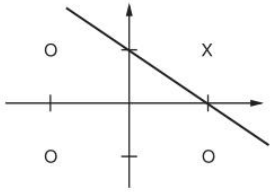
b) A set of training examples that is not linearly separable (i.e., that cannot be correctly classified by any straight line).

- X_1 and X_2 are the Perceptron inputs. Positive 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
$w_1 = 1$
$w_2 = 1$
$b = -1$
$q = 0$
$-1+x_1 + x_2 = 0$



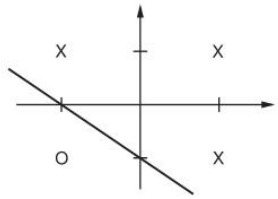
X : Class I (y = 1)
O : Class II (y = –1)

Fig. Q.20.1

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.20.2

Q.21 What is an Adaline ? Explain in detail.

Ans. :

- Adaline which stands for Adaptive Linear Neuron, is a network having a single linear unit.
- The basic structure of Adaline is similar to perceptron having an extra feedback loop with the help of which the actual output is compared with the desired/target output. After comparison on the basis of training algorithm, the weights and bias will be updated.
- Fig. Q.21.1 shows adaline.
- 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 $\theta \equiv w_0$

- 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

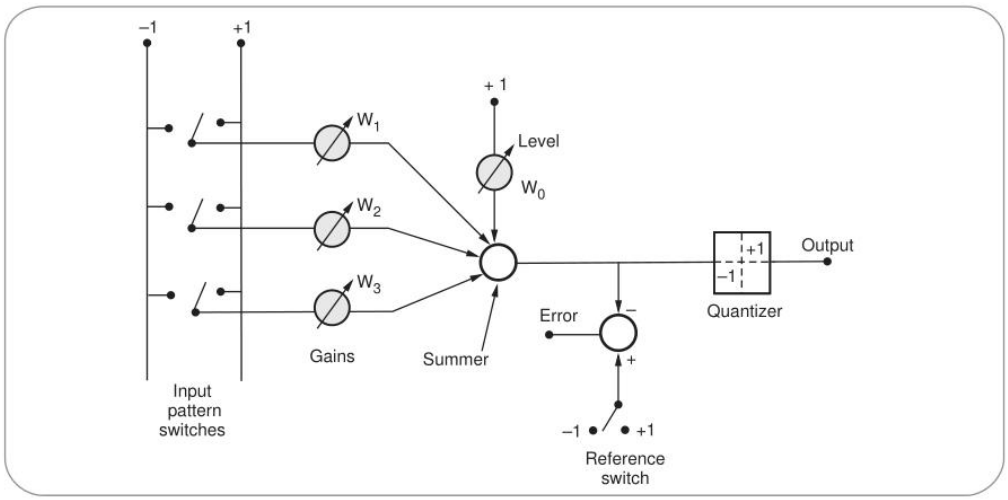


Fig. Q.21.1

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.
- 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 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

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

Q.22 What do you mean by linear separability ?

Ans. :

- Consider two-input patterns (X_1, X_2) being classified into two classes as shown in Fig. Q.22.1. 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.

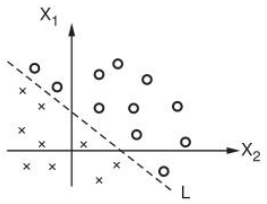


Fig. Q.22.1

- 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.
- 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 tow classes are said to be linearly inseparable.
- Linearly inseparable problems cannot be solved by the simple network, more sophisticated architecture is needed.

1.3 : Back-propagation Networks

Q.23 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.24 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.25 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.25.1 shows multilayer feed forward neural network.

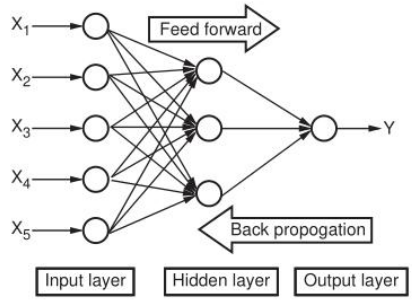


Fig. Q.25.1 Multilayer feed forward neural network

- 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_j would be labeled w_{hj} . The bias input node (x_0) is connected to all the hidden units, with weights w_{h0}.
- 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.26 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.27 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.28 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 :

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

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

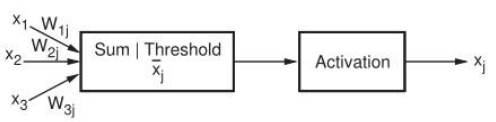


Fig. Q.28.1 Backpropagation MLP for node j

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.28.2 shows two layer back propagation MLP.

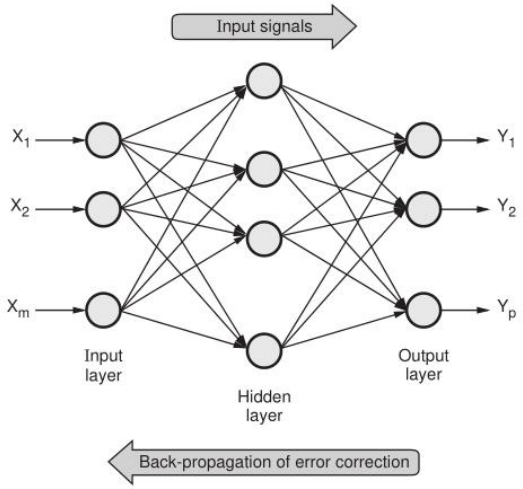


Fig. Q.28.2

- 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.29 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.30 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.

1.4 : Associative Memory Networks

Q.31 What is meant by associative memory ?

☞ [JNTU : May-17, Marks 2]

Ans. : An associative memory can be considered as a memory unit whose stored data can be identified for access by the content of the data itself rather than by an address or memory location. Associative memory is often referred to as Content Addressable Memory (CAM).

Q.32 Define auto associative memory.

Ans. : This is a single layer neural network in which the input training vector and the output target vectors are the same. The weights are determined so that the network stores a set of patterns. If vector "t" is the same as "s", the net is auto-associative.

Q.33 Describe auto-associative memory.

Ans. :

- An auto-associative net remembers one or more patterns. For an auto-associative net, the training input and target output vectors are identical.
- Auto-associative memories are content based memories which can recall a stored sequence when they are presented with a fragment or a noisy version of it.
- They are very effective in de-noising the input or removing interference from the input which makes them a promising first step in solving the cocktail party problem.
- The simplest version of auto-associative memory is linear associator which is a 2-layer feed-forward fully connected neural network where the output is constructed in a single feed-forward computation. The figure below illustrates its basic connectivity.

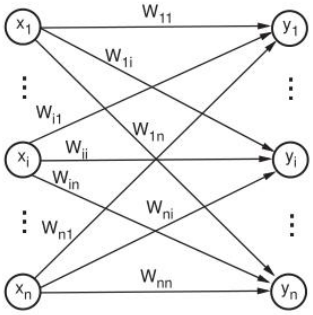


Fig. Q.33.1

- All inputs are connected to all outputs via the connection weight matrix W where W_{ij} denotes the strength of unidirectional connection from the i^{th} input to the j^{th} output.
- Since x = y in auto-associative memories, we have $x = W^T x$. Therefore, all stored sequences must be eigenvectors of matrix W. Assuming all stored sequences are orthogonal to each other, we can represent the weight matrix as $W = XX^{-1} = XX^T$ where T is orthonormal matrix of all stored sequences.
- The process of training is called storing the vectors. The representation can be bipolar or binary.

- Not only does it remember patterns exactly, but in addition, shown some degraded versions of a pattern, it returns the original uncorrupted version of the pattern.
- Either Hebb Rule, Delta Rule or Extended Delta Rule can be used for training.
- It is often the case that for auto-associative nets, the diagonal weights (those which connect an input component to the corresponding output component) are set to 0.

Q.34 What is Hebbian learning ?

☞[JNTU : May - 17, Marks 2]

Ans. : Hebb rule is the simplest and most common method of determining weights for an associative memory neural net. It can be used with patterns are represented as either binary or bipolar vectors.

Q.35 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.35.1 shows adaline.
- 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 $\theta \equiv w_0$

- 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.
- 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.

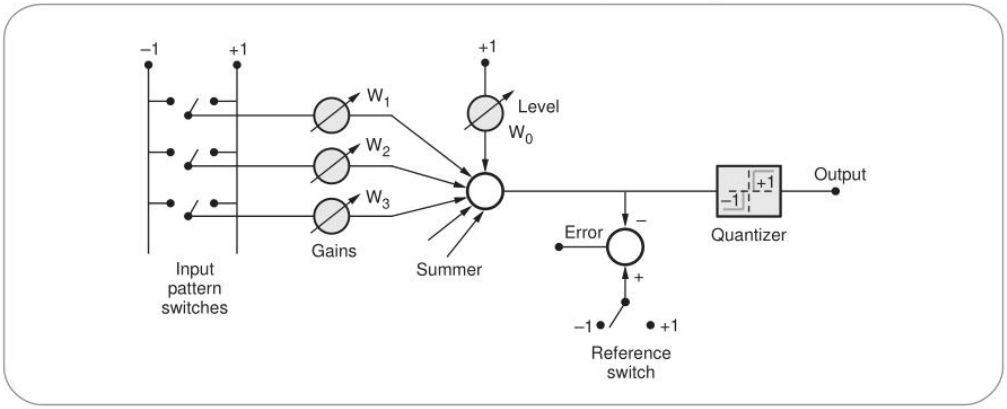


Fig. Q.35.1 Adaline

- 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.36 What is Bidirectional Associative Memory (BAM) ?

Ans. :

- The Hopfield network represents an **auto-associative** type of memory. It can retrieve a corrupted or incomplete memory but cannot associate this memory with another different memory.
- Human memory is essentially **associative**. One thing may remind us of another, and that of another, and so on. We use a chain of mental associations to recover a lost memory.
- If we forget where we left an umbrella, we try to recall where we last had it, what we were doing, and who we were talking to. We attempt to establish a chain of associations, and thereby to restore a lost memory
- **Bidirectional associative memory (BAM),** first proposed by **Bart Kosko**, is a hetero-associative network. It associates patterns from one set, set A, to patterns from another set, set B, and vice versa.
- Like a Hopfield network, the BAM can generalize and also produce correct outputs despite corrupted or incomplete inputs.
- Fig Q.36.1 shows BAM operation. (Fig. Q.36.1 shown on next page)
- The basic idea behind the BAM is to store pattern pairs so that when n-dimensional vector **X** from set A is presented as input, the BAM recalls m-dimensional vector **Y** from set B, but when **Y** is presented as input, the BAM recalls **X**.
- To develop the BAM, we need to create a correlation matrix for each pattern pair we want to store.
- The correlation matrix is the matrix product of the input vector **X**, and the transpose of the output vector $\mathbf{Y}^T$. The BAM weight matrix is the sum of all correlation matrices, that is

$$W = \sum_{m=1}^{M} X_m Y_m^T$$

where M is the number of pattern pairs to be stored in the BAM

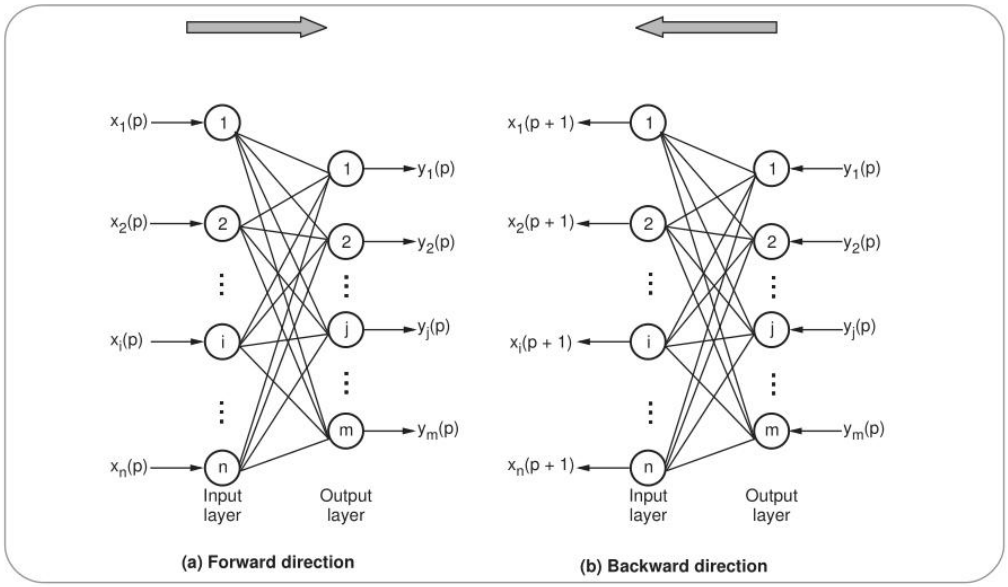


Fig. Q.36.1 Adaline

Q.37 Explain stability and storage capacity of the BAM.

Ans. : • The BAM is **unconditionally stable**. This means that any set of associations can be learned without risk of instability.

- The maximum number of associations to be stored in the BAM should not exceed the number of neurons in the smaller layer.
- The more serious problem with the BAM is incorrect **convergence**. The BAM may not always produce the closest association. In fact, a stable association may be only slightly related to the initial input vector.

Q.38 List the problems of BAM Network.

Ans. :

1. Storage capacity of the BAM: The maximum number of associations to be stored in the BAM should not exceed the number of neurons in the smaller layer.
2. Incorrect convergence. The BAM may not always produce the closest association

Q.39 What is content - addressable memory ?

Ans. : • A content-addressable memory is a type of memory that allows for the recall of data based on the degree of similarity between the input pattern and the patterns stored in memory.

- It refers to a memory organization in which the memory is accessed by its content as opposed to an explicit address like in the traditional computer memory system.
- Therefore, this type of memory allows the recall of information based on partial knowledge of its contents.

Q.40 What are the Delta Rule for Pattern Association ?

Ans. :

- When the input vectors are linearly independent, the Delta Rule produces exact solutions.
- Whether the input vectors are linearly independent or not, the Delta Rule produces a least squares solution, i.e., it optimizes for the lowest sum of least squared errors.

Q.41 What is continuous BAM ?

Ans. : Continuous BAM transforms input smoothly and continuously into output in the range [0, 1] using the logistic sigmoid function as the activation function for all units.

Q.42 Explain Hebbian rule with example.

Ans. : • In 1949, Donald Hebb proposed one of the key ideas in biological learning, commonly known as **Hebb's Law.** Hebb's Law states that if neuron i is near enough to excite neuron j and repeatedly participates in its activation, the synaptic connection between these two neurons is strengthened and neuron j becomes more sensitive to stimuli from neuron i.

- Hebb's Law can be represented in the form of two rules :
  1. If two neurons on either side of a connection are activated synchronously, then the weight of that connection is increased.
  2. If two neurons on either side of a connection are activated asynchronously, then the weight of that connection is decreased.
- Hebb's Law provides the basis for learning without a teacher. Learning here is a **local phenomenon** occurring without feedback from the environment.
- Using Hebb's Law we can express the adjustment applied to the weight w_{ij} at iteration p in the following form :

$$\Delta w_{ij}(p) = F[y_i(p), x_i(p)]$$

- As a special case, we can represent Hebb's Law as follows :

$$\Delta w_{ij}(p) = \alpha\, y_i(p)\, x_i(p)$$

where α is the learning rate parameter. This equation is referred to as the **activity product rule.**

- Hebbian learning implies that weights can only increase. To resolve this problem, we might impose a limit on the growth of synaptic weights. It can be done by introducing a non-linear **forgetting factor** into Hebb's Law :

$$\Delta w_{ij}(p) = \alpha\, y_i(p)\, x_i(p) - \varphi\, y_i(p)\, w_{ij}(p)$$

where φ is the forgetting factor.

- Forgetting factor usually falls in the interval between 0 and 1, typically between 0.01 and 0.1, to allow only a little "forgetting" while limiting the weight growth.

Hebbian learning algorithm

Step 1 : Initialisation

Set initial synaptic weights and thresholds to small random values, say in an interval [0, 1].

Step 2 : Activation

Compute the neuron output at iteration p

$$y_j(p) = \sum_{i=1}^{n} x_i(p)\, w_{ij}(p) - \theta_j$$

where n is the number of neuron inputs, and θ_j is the threshold value of neuron j.

Step 3 : Learning

Update the weights in the network :

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

where $w_{ij}(p)$ is the weight correction at iteration p. The weight correction is determined by the generalised activity product rule :

$$w_{ij}(p) = \varphi\, y_j(p)\, [\lambda\, x_i(p) - w_{ij}(p)]$$

Step 4 : Iteration

Increase iteration p by one, go back to Step 2.

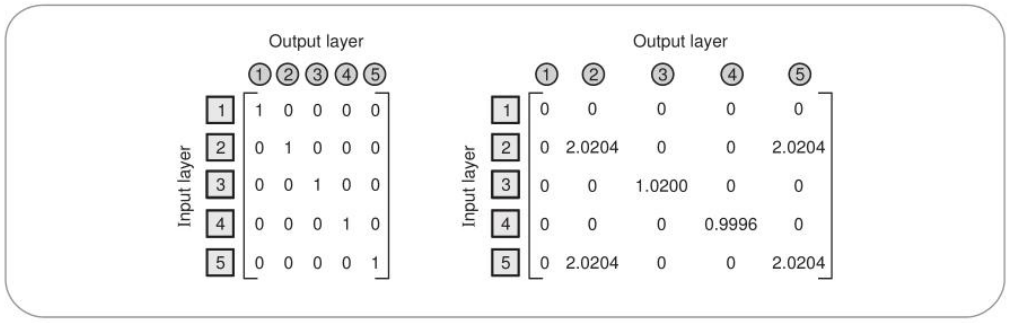
Hebbian learning example

- To illustrate Hebbian learning, consider a fully connected feed forward network with a single layer of five computation neurons. Each neuron is represented by a McCulloch and Pitts model with the sign activation function. The network is trained on the following set of input vectors :

$$x_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad x_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad x_3 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

$$x_4 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad x_5 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Initial and final weight matrices



- A test input vector, or probe, is defined as

$$X = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- When this probe is presented to the network, we obtain :

$$Y = \left\{ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2.0204 & 0 & 0 & 2.0204 \\ 0 & 0 & 1.0200 & 0 & 0 \\ 0 & 0 & 0 & 0.9996 & 0 \\ 0 & 2.0204 & 0 & 0 & 2.0204 \end{bmatrix} \right\} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} - \begin{bmatrix} 0.4940 \\ 0.2661 \\ 0.0907 \\ 0.9478 \\ 0.0737 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

1.5 : Hopfield Networks

Q.43 What is Discrete Hopfield Network ?

Ans. : A Hopfield network which operates in a discrete line fashion or in other words, it can be said the input and output patterns are discrete vector, which can be either binary (0,1) or bipolar (+1, – 1) in nature. The network has symmetrical weights with no self-connections i.e., $w_{ij} = w_{ji}$ and $w_{ii} = 0$.

Q.44 What is Continuous Hopfield Network ?

Ans. : In comparison with Discrete Hopfield network, continuous network has time as a continuous variable. It is also used in auto association and optimization problems such as travelling salesman problem

Q.45 Explain what is meant by capacity of Hopfield network.

Ans. : • Network capacity of the Hopfield network model is determined by neuron amounts and connections within a given network. Therefore, the number of memories that are able to be stored is dependent on neurons and connections.

- Furthermore, it was shown that the recall accuracy between vectors and nodes was 0.138. Therefore, it is evident that many mistakes will occur if one tries to store a large number of vectors.

- When the Hopfield model does not recall the right pattern, it is possible that an intrusion has taken place, since semantically related items tend to confuse the individual, and recollection of the wrong pattern occurs.
- Therefore if we store p patterns in a Hopfield network with a large number of N nodes, then the probability of error, i.e., the probability that $C_i^k > 1$ is

$$P_{error} = P\,(C_i^k > 1) \approx \frac{1}{\sqrt{2\,\pi\,\sigma}} \int_1^{\infty} \exp(-x^2/2\,\sigma^2)\,dx$$

$$= \frac{1}{2}(1 - \text{erf}(1/\sqrt{2\sigma^2}))$$

$$= \frac{1}{2}(1 - \text{erf}(\sqrt{N/2\,p}))$$

where the error function erf is given by :

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-s^2)\,ds$$

Therefore, given N and p we can find out the probability

P_{error} of error for a single neuron of a stored pattern.

Q.46 Explain with neat diagram the architecture of Hopfield neural network.

Ans. : • The **Hopfield** model is a single-layered recurrent network. Like the associative memory, it is usually initialized with appropriate weights instead of being trained.

- Hopfield Neural Network (HNN) is a model of auto-associative memory. It is a single layer neural network with feedbacks. Fig. Q.46.1 shows Hopfield network of three units. The Hopfield network is created by supplying input data vectors, or pattern vectors, corresponding to the different classes. These patterns are called class patterns.

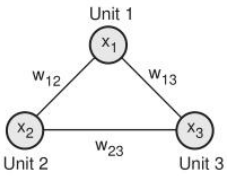


Fig. Q.46.1 Hopfield network of three units

- Hopfield model consists of a single layer of processing elements where each unit is connected to every other unit in the network other than itself.
- The output of each neuron is a binary number in {–1,1}. The output vector is the state vector. Starting from an initial state (given as the input vector), the state of the network changes from one to another like an automaton. If the state converges, the point to which it converges is called the attractor.
- In its simplest form, the output function is the sign function, which yields 1 for arguments ≥ 0 and –1 otherwise.
- The connection weight matrix W of this type of network is square and symmetric. The units in the Hopfield model act as both input and output units.
- A Hopfield network consists of "n" totally coupled units. Each unit is connected to all other units except itself. The network is symmetric because the weight w_{ij} for the connection between unit i and unit j is equal to the weight w_{ij} of the connection from unit j to unit i. The absence of a connection from each unit to itself avoids a permanent feedback of its own state value.
- Hopfield networks are typically used for classification problems with binary pattern vectors.
- Hopfield model is classified into two categories :

1. Discrete Hopfield Model
2. Continuous Hopfield Model

- In both discrete and continuous Hopfield network weights trained in a one-shot fashion and not trained incrementally as was done in case of Perceptron and MLP.
- In the discrete Hopfield model, the units use a slightly modified bipolar output function where the states of the units, i.e., the output of the units remain the same if the current state is equal to some threshold value.
- The continuous Hopfield model is just a generalization of the discrete case. Here, the units use a continuous output function such as the sigmoid or hyperbolic tangent function. In the continuous Hopfield model, each unit has an associated capacitor C_i and resistance r_i that model the capacitance and resistance of real neuron's cell membrane, respectively.

Q.47 Give the limitation and application of Hopfield networks. ☞[JNTU : June-13, Marks 7]

Ans. : Limitations of Hopfield Networks

- Severely limited in the number of patterns p that can be stored reliably in N nodes.
- Generally, the maximum number of patterns must be below 0.15N for reliable performance;
- Avalanche in error occurs above 0.138N.
- Too many patterns will result in spurious outputs; i.e., outputs not corresponding to any stored pattern.
- Storing similar patterns can cause errors in output.

Application:

- Common applications are those where pattern recognition is useful, and Hopfield networks have been used for image detection and recognition, enhancement of X-Ray images, medical image restoration, etc.
- The Hopfield network is commonly used for auto-association and optimization tasks.

Q.48 Why Hopfield neural network is an associative memory ?

Ans. :

- Starting from any initial state, the HNN will change its state until the energy function approaches to the minimum.
- The minimum point is called the attractor.
- Patterns can be stored in the network in the form of attractors.
- The initial state is given as the input, and the state after convergence is the output.

Q.49 Explain applications of the Hopfield Network.

Ans. :

1. The Hopfield network can be used as an effective interface between analog and digital devices, where the input signals to the network are analog and the output signals are discrete values.
2. Associative memory is a major application of the Hopfield network.
3. The energy minimization ability of the Hopfield network is used to solve optimization problems.
4. The various applications of Hopfield networks are as given below. Hopfield network remembers cues from the past and does not complicate the training procedure.
5. The Hopfield network can be used for converting analog signals into the digital format, for associative memory.

Q.50 What is associative memory ?

Ans. : Associative memory neural nets are single layer nets in which the weights are determined in such a way that net can store a set of pattern associations.

Q.51 Explain relation between BAM and Hopfield nets.

Ans. : • Discrete Hopfield net and the BAM net are closely related.

- The Hopfield net can be viewed as an auto-associative BAM with the X-layer and Y-layer treated as a single layer and the diagonal of the symmetric weight matrix set to zero.
- BAM can be viewed as a special case of a Hopfield net which contains all of the X-layer and Y-layer neurons, but with no interconnections between two X-layer neurons or between two Y-layer neurons.
- X-layer neurons must be update their activations before any of the Y-layer neurons update theirs; then all Y field neuron update before the next round of X-layer updates.
- The update of the neurons with the X-layer or within the Y-layer can be done at the same time because a change in the activation of an X-layer neuron does not affect the net input to any other X-layer unit and similarly for the Y-layer units.

Q.52 Explain Hopfield model and apply it to traveling salesman problem.

☞[JNTU : May-17, Marks 5]

Ans. :

- Figure Q.52.1 shows a TSP defined over a transportation network. The artificial neural network encoding that problem is also shown.

- In the transportation network, the five vertices stand for cities and the links are labeled or weighted by the inter-city distances dij.
- A feasible solution to that problem is the tour Montreal-Boston-NY-LA-Toronto-Montreal, as shown by the bold arcs.
- In the TSP context, the weights are derived in part from the inter-city distances. They are chosen to penalize infeasible tours and, among the feasible tours, to favor the shorter ones.
- The weight on the connection between the units that represent a visit to cities.
- Consequently, that connection should be inhibitory (negative weight), because two cities cannot occupy the same exact position.
- The first unit to be activated will inhibit the other unit via that connection, so as to prevent an infeasible solution to occur.

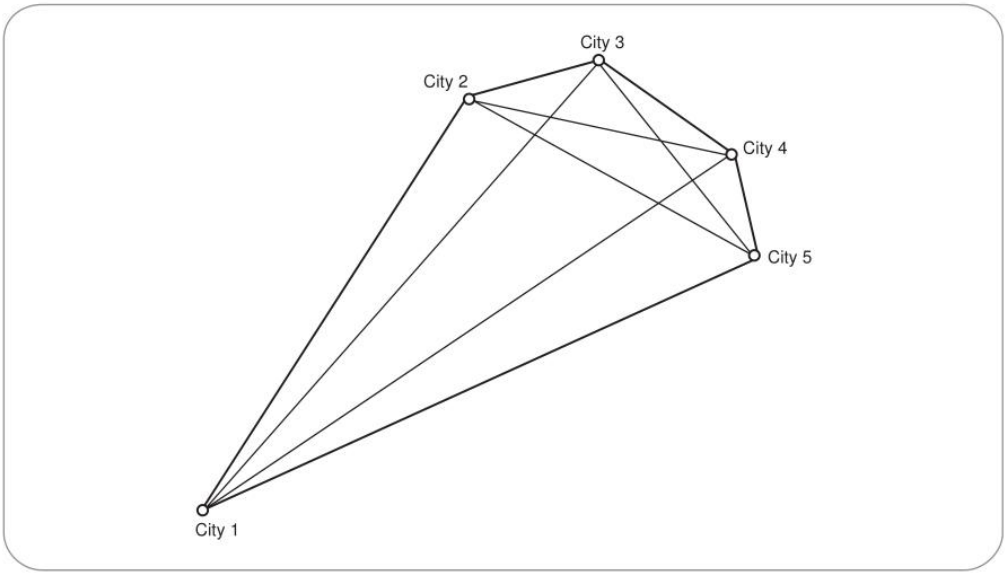


Fig. Q.52.1 Schematic of biological neuron

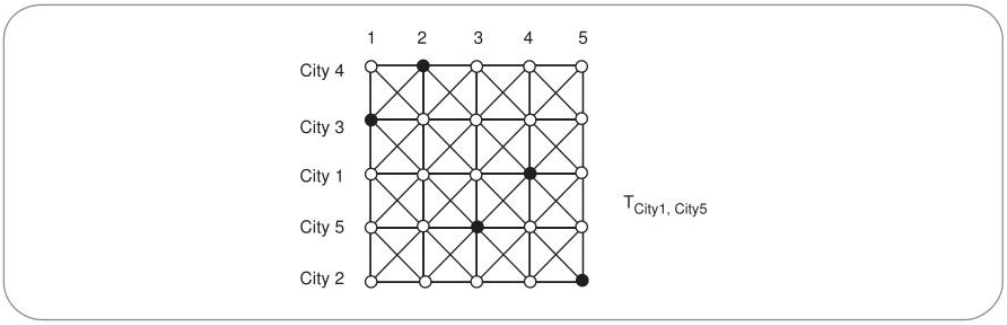


Fig. Q.52.2 Schematic of biological neuron

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]**

Q.10 A _____ is an information-processing unit that is fundamental to the operation of a neural network.

Q.11 An _____ is a single-layer network in which the weights are determined in such a way that the network can store a set of pattern associations.

Q.12 Auto-associative memories are _____ memories which can recall a stored sequence.

Q.13 _____ are elementary functional and structural units that mediate the interaction between neurons. ☞ **[JNTU : Feb. - 17]**

Q.14 In neural network terminology a threshold function is commonly known as _____. ☞ **[JNTU : Feb. - 17]**

Q.15 An error correction learning algorithm which is applied to a multilayer feed forward network leads to _____ type of problem . ☞ **[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

Q.6 A possible neuron specification to solve the AND problem requires a minimum of _____.

- a single neuron
- b two neurons

c three neurons d four neurons

Q.7 A perceptron takes a vector of real-valued inputs, calculates a linear combination of these inputs, then outputs.

☞ [JNTU : Aug.-16, Feb.-17]

a 1 or –1 b 0 or 1

c –1 or 0 d none

Q.8 The_____algorithm computes the version space containing all hypotheses from H that are consistent with an observed sequence of training examples. ☞ [JNTU : Feb.-17]

a inductive Hypothesis

b artificial Neural Network

c candidate Elimination

d none

Q.9 If the training examples are not linearly separable, the delta rule converges toward a approximation to the target concept.

☞ [JNTU : Feb.-17]

a Over fit b Under fit

c Doesn't fit d Best fit

Q.10 What is shape of dendrites like _____

a oval b round

c tree d rectangular

Q.11 What is the objective of BAM ?

a to store pattern pairs

b to recall pattern pairs

c to store a set of pattern pairs and they can be recalled by giving either of pattern as input

d none of the mentioned

Q.12 Hetero-associative memory is also known as ?

a unidirectional memory

b bidirectional memory

c multidirectional associative memory

d temporal associative memory

Q.13 What are the general tasks that are performed with backpropagation algorithm ?

a pattern mapping

b function approximation

c prediction

d all of the above

Q.14 Adaline which stands for _____.

a adaptive Linear Neuron

b address Linear Neuron

c adaptive Linear Network

d adaptive Neural Neuron

Q.15 BAM stands for _____.

a Bidirectional Adaptive memory

b Backpropagation associative memory

c Bidirectional associative memory

d Bidirectional associative machine

Q.16 The Hopfield network consists of a set of neurons forming a multiple loop _____ system.

a unidirectional b parallel

c feedback d feedforward

Q.17 A _____ Hopfield net can be used to determine whether an input vector is a known vector or an unknown vector.

a binary b discrete

c autoassociative d All

Q.18 Neural networks are complex _____ with many parameters.

a linear functions

b nonliner functions

c discrete functions

d exponential functions

Q.19 The network that inolves backward links from output to the input and hidden layers is called as _____.

a self organizing maps

b perceptrons

c recurrent neural network

d multi layered perceptron

Q.20 A Neuro software is :

a a software used to analyze neurons

b it is powerful and easy neural network.

c designed to aid experts in real word

d it is software used by Neuro surgeon.

Q.21 A neuron can send at a time how many signals ?

a Multiple b One

c None d 10^{10}

Q.22 _____ belongs to the class of single layer feed forward or recurrent network architecture depending on its association capability.

a Fuzzy systems

b Associative memory

c Pattern classification

d Genetic algorithms.

Q.23 The BAM (Bidirectional Associative Memory) significantly belongs to _____.

a Auto associative

b Fuzzy associative

c Genetic associative

d Hetero associative.

Answer Key for Fill in the Blanks

Q.1	artificial neurons	**Q.2**	activation
Q.3	multi layer	**Q.4**	perceptron
Q.5	learning	**Q.6**	rule post pruning
Q.7	Neural network	**Q.8**	boolean
Q.9	training	**Q.10**	neuron
Q.11	associative network	**Q.12**	content based
Q.13	central nervous system	**Q.14**	activation function
Q.15	supervised learning		

Answer Key for Multiple Choice Questions

Q.1	a	**Q.2**	c	**Q.3**	d	**Q.4**	a
Q.5	c	**Q.6**	a	**Q.7**	a	**Q.8**	c
Q.9	d	**Q.10**	c	**Q.11**	c	**Q.12**	b
Q.13	d	**Q.14**	a	**Q.15**	c	**Q.16**	c
Q.17	a	**Q.18**	a	**Q.19**	c	**Q.20**	b
Q.21	a	**Q.22**	b	**Q.23**	d		

END...

UNIT - II

2 Unsupervised Learning Network

2.1 : Introduction

Q.1 What is unsupervised learning ?

Ans. : In an unsupervised learning, the network adapts purely in response to its inputs. Such networks can learn to pick out structure in their input.

Q.2 Explain unsupervised learning.

Ans. : Unsupervised learning : • The model is not provided with the correct results during the training. It can be used to cluster the input data in classes on the basis of their statistical properties only. Cluster significance and labeling.

- The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes. All similar inputs patterns are grouped together as clusters.
- If matching pattern is not found, a new cluster is formed. There is no error feedback.
- External teacher is not used and is based upon only local information. It is also referred to as **self-organization.**
- They are called unsupervised because they do not need a teacher or super-visor to label a set of training examples. Only the original data is required to start the analysis.
- In contrast to supervised learning, unsupervised or self-organized learning does not require an external teacher. During the training session, the neural network receives a number of different input patterns, discovers significant features in these patterns and learns how to classify input data into appropriate categories.
- Unsupervised learning algorithms aim to learn rapidly and can be used in real-time. Unsupervised learning is frequently employed for data clustering, feature extraction etc.
- Another mode of learning called recording learning by Zurada is typically employed for associative memory networks. An associative memory networks is designed by recording several idea patterns into the networks stable states.

Q.3 What is semi-supervised learning ?

Ans. : Semi-supervised Learning : • Semi-supervised learning uses both labeled and unlabeled data to improve supervised learning. The goal is to learn a predictor that predicts future test data better than the predictor learned from the labeled training data alone.

- Semi-supervised learning is motivated by its practical value in learning faster, better, and cheaper.
- In many real world applications, it is relatively easy to acquire a large amount of unlabeled data x.
- For example, documents can be crawled from the Web, images can be obtained from surveillance cameras, and speech can be collected from broadcast. However, their corresponding labels y for the

prediction task, such as sentiment orientation, intrusion detection, and phonetic transcript, often requires slow human annotation and expensive laboratory experiments.

- In many practical learning domains, there is a large supply of unlabeled data but limited labeled data, which can be expensive to generate. For example : text processing, video-indexing, bioinformatics etc.
- Semi-supervised Learning makes use of both labeled and unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. When unlabeled data is used in conjunction with a small amount of labeled data, it can produce considerable improvement in learning accuracy.
- Semi-supervised learning sometimes enables predictive model testing at reduced cost.
- **Semi-supervised classification :** Training on labeled data exploits additional unlabeled data, frequently resulting in a more accurate classifier.
- **Semi-supervised clustering :** Uses small amount of labeled data to aid and bias the clustering of unlabeled data.

Q.4 Explain difference between supervised and unsupervised learning.

Ans. :

Sr. No.	Supervised Learning	Unsupervised Learning
1.	Desired output is given.	Desired output is not given.
2.	It is not possible to learn larger and more complex models than with supervised learning	It is possible to learn larger and more complex models with unsupervised learning
3.	Use training data to infer model	No training data is used
4.	Every input pattern that is used to train the network is associated with an output pattern.	The target output is not presented to the network.
5.	Trying to predict a function from labeled data	Try to detect interesting relations in data.
6.	Supervised learning requires that the target variable is well defined and that a sufficient number of its values are given.	For unsupervised learning typically either the target variable is unknown or has only been recorded for too small a number of cases.
7.	Example : Optical character recognition	Example : Find a face in an image.
8.	We can test our model	We can not test our model
9.	Supervised learning is also called classification	Unsupervised learning is also called clustering.

Q.5 What is winner-take-all learning network ?

Ans. : • Most unsupervised neural networks rely on algorithms to compute and compare distances, determine the winner node with the highest level of activation and use these to adapt network weights

- Among all competing nodes, only one will win and all others will lose. We mainly deal with single winner WTA, but multiple winners WTA are possible.
- Easiest way to realize WTA : have an external, central arbitrator to decide the winner by comparing the current outputs of the competitors. This is biologically unsound.

Q.6 List the types of fixed weight competitive nets.

Ans. : Fixed weight competitive nets are Maxnet, Mexican Hat and Hamming Net.

Q.7 What is hamming net ?

Ans. : Hamming net finds the similarities between the input pattern and the weight vectors of all neurons.

Q.8 What is hamming distance ?

Ans. : The number of different bits in two binary or bipolar vectors X_1 and X_2 is called the hamming distance between the vectors and is denoted by $H[X_1, X_2]$.

Q.9 Explain hamming net with diagram.

Ans. : • Hamming net is a single layer neural network.

- Hamming network is a neural network mode that is specifically designed to address the pattern recognition with inputs from neural network in a bipolar form.

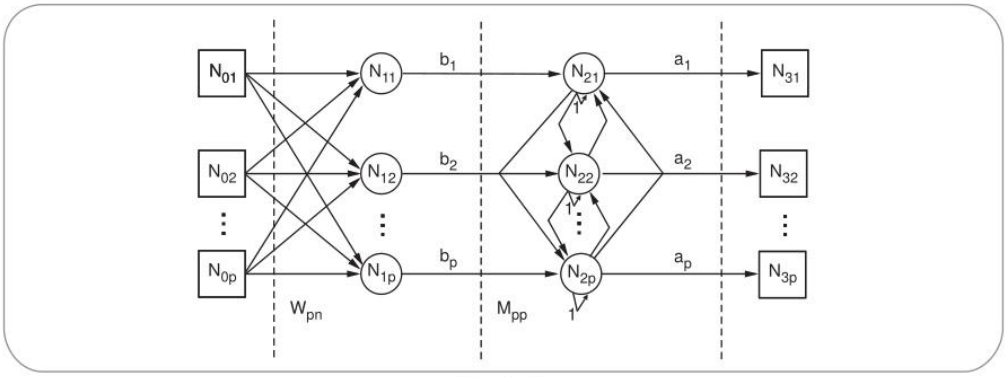


Fig. Q.9.1

- In the process, hamming network use the hamming distance as a similarity indicator between two vectors, and Maxnet serves as a subnet to determine the unit that has the biggest net input
- It is the neural network implementation of the hamming distance based nearest neighbor classifier.
- The inputs are binary numbers {0,1} or {– 1,1}.
- We consider only bi-polar case here. The outputs are the similarities between the input pattern and the weight vectors of the neurons.
- The number of inputs is n, which is the dimensionality of the pattern space. The number of outputs is p, which is the number of patterns to store.
- The hamming net uses MAXNET as a subnet to find the unit with the largest net input
- Input layer neurons are programmed to identify a fixed number of patterns; the number of neurons in this layer matches the number of those patterns (M neurons - M patterns).
- Outputs of these neurons realise the function, which "measures" the similarity of an input signal to a given pattern.
- The output layer is responsible for choosing the pattern, which is the most similar to testing signal. In this layer, the neuron with the strongest response stops the other neurons responses.

Q.10 Define maxnet.

Ans. : Maxnet is one of the artificial neural networks based on competition. Generally, it is used for pattern identification. Maxnet can be used by another model of artificial neural network such as hamming network to get neuron with the biggest input.

Q.11 Write about Maxican Hat network.

Ans. : • The Mexican Hat network is a more general contrast enhancing subnet than the MAXNET. Three types of links can be found in such network :

1. Each neuron is connected with excitatory (positively weighted) links to a number of "cooperative neighbors," neurons that are in close proximity.
2. Each neuron is also connected with inhibitory links (with negative weights) to a number of "competitive neighbors," neurons that are somewhat further away.
3. There may also be a number of neurons, further away still, to which the neuron is not connected.

• All of these connections are within a particular layer of a neural net, so, as in the case of MAXNET, the neurons receive an external signal in addition to these interconnection signals

2.2 : Kohonen Self-Organizing Feature Maps

Q.12 What are the goals of competitive learning.

Ans. :

1. Learn to form classes/clusters of examples/sample patterns according to similarities of these examples.
2. Patterns in a cluster would have similar features.
3. No prior knowledge as what features are important for classification, and how many classes are there.

Q.13 Define self - organizing map.

Ans. : The self-organizing map is one of the most popular neural network models. It belongs to the category of competitive learning networks. The self-organizing map is based on unsupervised learning, which means that no human intervention is needed during the learning and that little needs to be known about the characteristics of the input data.

Q.14 What is principle goal of the self-organizing map ?

Ans. : The principal goal of the Self-Organizing Map (SOM) is to transform an incoming signal pattern of arbitrary dimension into a one - or two - dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion.

Q.15 Explain components of self organization.

Ans. : • The self-organization process involves four major components :

1. **Initialization :** All the connection weights are initialized with small random values.
2. **Competition :** For each input pattern, the neurons compute their respective values of a discriminant function which provides the basis for competition. The particular neuron with the smallest value of the discriminant function is declared the winner.
3. **Cooperation :** The winning neuron determines the spatial location of a topological neighbourhood of excited neurons, thereby providing the basis for cooperation among neighbouring neurons.
4. **Adaptation :** The excited neurons decrease their individual values of the discriminant function in relation to the input pattern through suitable adjustment of the associated connection weights, such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.

Q.16 List the stages of the SOM algorithm.

Ans. :

1. Initialization - Choose random values for the initial weight vectors w_j.
2. Sampling - Draw a sample training input vector x from the input space.
3. Matching - Find the winning neuron I(x) with weight vector closest to input vector.
4. Updating - Apply the weight update equation
$$\Delta w_{ji} = \eta(t)\ T_{j,I(x)}(t)\ (x_i - w_{ji})$$
5. Continuation - Keep returning to step 2 until the feature map stops changing.

Q.17 Explain an essential ingredients and parameters of the SOM algorithm.

Ans. : An essential ingredients and parameters of the SOM algorithm are as follows :

1. Continuous input space of activation patterns that are generated in accordance with a certain probability distribution.

2. Topology of the network in the form of a lattice of neurons, which defines a discrete output space;
3. Time-varying neighborhood function h_j, i(**x**)(n) that is defined around a winning neuron i(**x**);
4. Learning-rate parameter that starts at an initial value and then decreases gradually with time , but never goes to zero.

Q.18 What is self-organizing feature maps ? List and explain its components.

Ans. : • In competitive learning, neurons compete among themselves to be activated. While in Hebbian learning, several output neurons can be activated simultaneously, in competitive learning, only a single output neuron is active at any time. The output neuron that wins the "competition" is called the winner-takes-all neuron.

- Such competition can be implemented by having lateral inhibition connections between the neurons. The result is that the neurons are forced to organise themselves. For obvious reasons, such a network is called a Self Organizing Map (SOM).
- The Self-Organizing Map is based on unsupervised learning, which means that no human intervention is needed during the learning and that little needs to be known about the characteristics of the input data.
- It provides a topology preserving mapping from the high dimensional space to map units. Map units or neurons, usually form a two-dimensional lattice and thus the mapping is a mapping from high dimensional space onto a plane.
- Brain is a self-organizing system that can learn by itself by changing (adding, removing, strengthening) the interconnections between neurons. Formation of feature maps in the brain that have a linear or planar topology.
- Use the SOM for clustering data without knowing the class memberships of the input data. The SOM can be used to detect features inherent to the problem and thus has also been called the **Self-Organizing Feature Map**.
- The property of topology preserving means that the mapping preserves the relative distance between the points. Points that are near each other in the input space are mapped to nearby map units in the SOM. The SOM can thus serve as a cluster analyzing tool of high-dimensional data. Also, the SOM has the capability to generalize.
- Generalization capability means that the network can recognize or characterize inputs it has never encountered before. A new input is assimilated with the map unit it is mapped to. Fig. Q.18.1 shows organization of the mapping.
- Here points x in the input space mapping to points I(x) in the output space. Each point "I" in the output space will map to a corresponding point w(I) in the input space.

Components of Self Organization : Refer Q.15.

Q.19 Write short note on Kohonen self organizing.

Ans. : • Kohonen self organizing networks are also called **Kohonen features** maps or **topology preserving maps** are used to solve competition based network paradigm for data clustering.

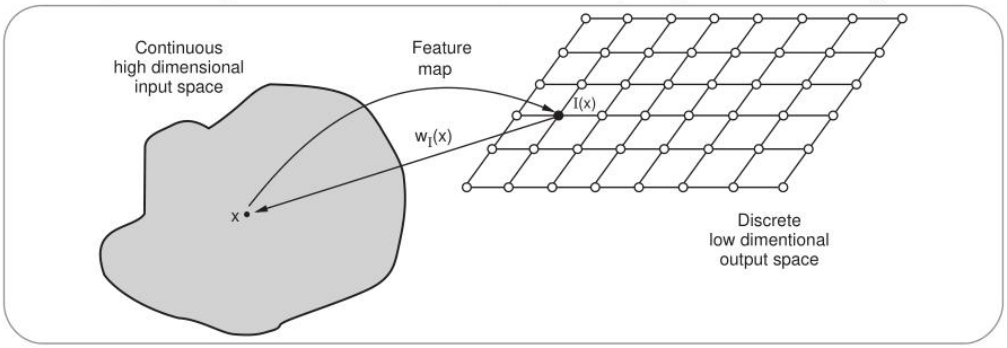


Fig. Q.18.1 Organization of the mapping

- The Kohonen model provides a topological mapping. It places a fixed number of input patterns from the input layer into a higher-dimensional output or Kohonen layer.
- Training in the Kohonen network begins with the winner's neighbourhood of a fairly large size. Then, as training proceeds, the neighbourhood size gradually decreases.
- Fig. Q.19.1 shows a simple Kohonen self organizing network with 2 inputs and 49 outputs. The learning feature map is similar to that of competitive learning networks.
- A similarity measure is selected and the winning unit is considered to be the one with the largest activation. For this **Kohonen features** maps all the weights in a neighborhood around the winning units are also updated. The neighborhood's size generally decreases slowly with each iteration.
- Step for how to train a Kohonen self organizing network is as follows :

For n-dimensional input space and m output neurons :

1. Choose random weight vector w_i for neuron i, i = 1, ..., m.
2. Choose random input x.
3. Determine winner neuron k : $||w_k - x| = \min_i ||w_i - x||$ (Euclidean distance)
4. Update all weight vectors of all neurons i in the neighborhood of neuron k : $w_i := w_i + \eta \cdot \varphi(i, k) \cdot (x - w_i)$ (w_i is shifted towards x)
5. If convergence criterion met, STOP. Otherwise, narrow neighborhood function and learning parameter η and go to (2).

Competitive learning in the Kohonen network

- To illustrate competitive learning, consider the Kohonen network with 100 neurons arranged in the form of a two-dimensional lattice with 10 rows and 10 columns. The network is required to classify two-dimensional input vectors - each neuron in the network should respond only to the input vectors occurring in its region.
- The network is trained with 1000 two-dimensional input vectors generated randomly in a square region in the interval between –1 and +1. The learning rate parameter a is equal to 0.1.

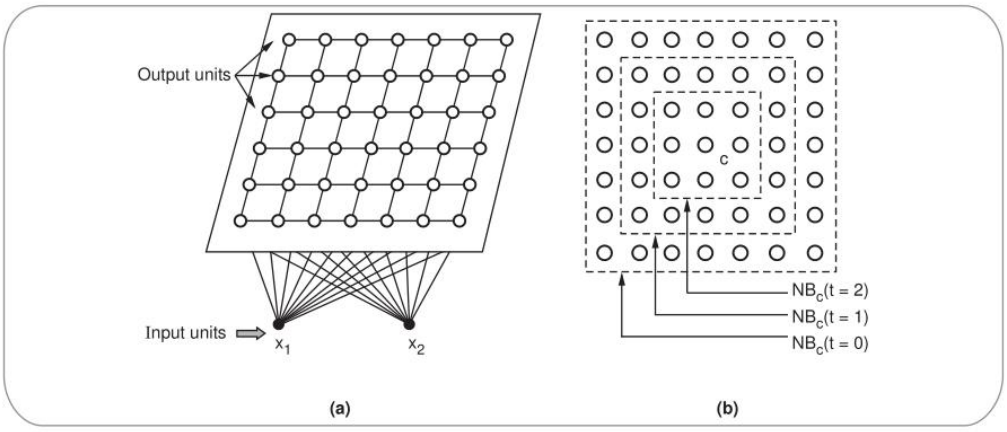


Fig. Q.19.1 Simple Kohonen self organizing network

2.3 Learning Vector Quantization

Q.20 Write short note on learning vector quantization.

Ans. : • Learning Vector Quantization (LVQ) is adaptive data classification method. It is based on training data with desired class information.

- LVQ uses unsupervised data clustering techniques to preprocesses the data set and obtain cluster centers.
- Fig. Q.20.1 shows the network representation of LVQ.

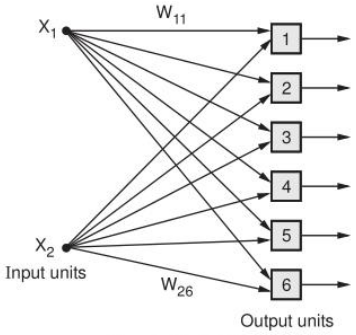


Fig. Q.20.1 LVQ

- Here input dimension is 2 and the input space is divided into six clusters. The first two clusters belong to class 1, while other four clusters belong to class 2.
- THE LVQ learning algorithm involves two steps :
  1. An unsupervised learning data clustering method is used to locate several cluster centers without using the class information.
  2. The class information is used to fine tune the cluster centers to minimize the number of misclassified cases.
- The number of cluster can either be specified a priori or determined via a cluster technique capable of adaptively adding new clusters when necessary. Once the clusters are obtained, their classes must be labeled before moving to second step. Such labeling is a achieved by **voting method**.

Learning method :

- The weight vector (w) that is closest to the input vector (x) must be found. If **x** belongs to the same class, we move **w** towards **x**; otherwise we move w away from the input vector **x**.

Step 1 : Initialize the cluster centers by a clustering method.

Step 2 : Label each cluster by the voting method.

Step 3 : Randomly select a training input vector **x** and find k such that $|| x - w_k ||$ is a minimum.

Step 4 : If x and w_k belongs to the same class, update w_k by

$$\Delta W_k = N (x - W_k)$$

Otherwise update w_k by

$$\Delta W_k = -\eta (X - W_k)$$

Step 5 : If the maximum number of iterations is reached, stop. Otherwise return to step 3.

2.4 : Counter Propagation Networks

Q.21 What is counter-propagation network ?

Ans. : Counter-propagation networks are multilayer networks based on a combination of input, clustering and output layers. It can be used to compress data, to approximate functions or to associate patterns.

Q.22 How does counter - propagation nets are trained ?

Ans. : • Counter-propagation nets are trained in two stages :

1. **First stage :** The input vectors are clustered. The clusters that are formed may be based on either the dot product metric or the Euclidean norm metric.
2. **Second stage :** The weights from the cluster units to the output units are adapted to produce the desired response.

Q.23 List the possible drawback of counter-propagation networks.

Ans. : • Training a counter-propagation network has the same difficulty associated with training a Kohonen network.

- Counter-propagation networks tend to be larger than backpropagation networks. If a certain number of mappings are to be learned, the middle layer must have that many number of neurons.

Q.24 Explain full Counter-Propagation Network (CPN).

Ans. : • The full CPN allows to produce a correct output even when it is given an input vector that is partially incomplete or incorrect.

- Full counter-propagation was developed to provide an efficient method of representing a large number of vector pairs, x : y by adaptively constructing a lookup table.
- It produces an approximation x* : y* based on input of an x vector or input of a y vector only, or input of an x : y pair, possibly with some distorted or missing elements in either or both vectors.
- In first phase, the training vector pairs are used to form clusters using either dot product or Euclidean distance. If dot product is used, normalization is a must.
- This phase of training is called as in star modeled training. The active units here are the units in the x-input, z-cluster and y-input layers. The winning unit uses standard Kohonen learning rule for its weigh updation.
- During second phase, the weights are adjusted between the cluster units and output units.
- In this phase, we can find only the J unit remaining active in the cluster layer.
- The weights from the winning cluster unit J to the output units are adjusted, so that vector of activation of units in the y output layer, y*, is approximation of input vector y; and x* is an approximation of input vector x.
- The architecture of CPN resembles an instar and outstar model.
- The model which connects the input layers to the hidden layer is called Instar model and the model which connects the hidden layer to the output layer is called Outstar model.
- The weights are updated in both the Instar (in first phase) and Outstar model (second phase).
- The network is fully interconnected network.

Q.25 How forward-only differs form full counter-propagation nets ?

Ans. : • In full counter-propagation, only the x vectors to form the clusters on the Kohonen units during the first stage of training.

- The original presentation of forward-only counter-propagation used the Euclidean distance between the input vector and the weight vector for the Kohonen unit.

Q.26 What is forward only counter-propagation ?

Ans. : • Is a simplified version of the full counterpropagation.

- Are intended to approximate y = f(x) function that is not necessarily invertible.
- It may be used if the mapping from x to y is well defined, but the mapping from y to x is not.

2.5 : Adaptive Resonance Theory

Q.27 Define plasticity.

Ans. : The ability of a net to respond to learn a new pattern equally well at any stage of learning is called plasticity.

Q.28 List the components of ART1.

Ans. : Components are as follows :

1. The short term memory layer (F1)
2. The recognition layer (F2) : It contains the long term memory of the system.
3. Vigilance Parameter (ρ) : A parameter that controls the generality of the memory. Larger ρ means more detailed memories, smaller ρ produces more general memories.

Q.29 Draw and explain the architecture of ART. What is its use and types ?

Ans. : Gail Carpenter and Stephen Grossberg (Boston University) developed the Adaptive Resonance learning model. How can a system retain its previously learned knowledge while incorporating new information.

- Adaptive resonance architectures are artificial neural networks that are capable of stable categorization of an arbitrary sequence of unlabeled input patterns in real time. These architectures are capable of continuous training with non-stationary inputs.

- Some models of Adaptive Resonance Theory are :
  1. ART1 - Discrete input.
  2. ART2 - Continuous input.
  3. ARTMAP - Using two input vectors, transforms the unsupervised ART model into a supervised one.
- Various others : Fuzzy ART, Fuzzy ARTMAP (FARTMAP), etc…
- The primary intuition behind the ART model is that object identification and recognition generally occur as a result of the interaction of 'top-down' observer expectations with 'bottom-up' sensory information.
- The basic ART system is an **unsupervised learning model.** It typically consists of a comparison field and a recognition field composed of neurons, a vigilance parameter, and a reset module. However, ART networks are able to grow additional neurons if a new input cannot be categorized appropriately with the existing neurons.
- ART networks tackle the stability-plasticity dilemma :
  1. **Plasticity :** They can always adapt to unknown inputs if the given input cannot be classified by existing clusters.
  2. **Stability :** Existing clusters are not deleted by the introduction of new inputs.
  3. **Problem :** Clusters are of fixed size, depending on ρ.
- Fig. Q.29.1 shows ART-1 Network.

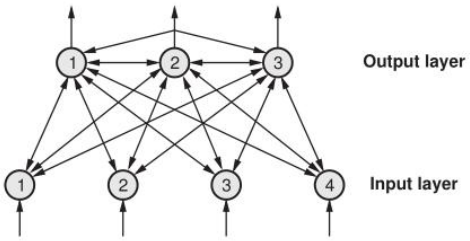


Fig. Q.29.1 ART 1 network

- ART-1 networks, which receive binary input vectors. Bottom-up weights are used to determine output-layer candidates that may best match the current input.
- Top-down weights represent the "prototype" for the cluster defined by each output neuron. A close match between input and prototype is necessary for categorizing the input.
- Finding this match can require multiple signal exchanges between the two layers in both directions until "resonance" is established or a new neuron is added.
- The basic ART model, ART1, is comprised of the following components :
  1. The short term memory layer : F1 - Short term memory.
  2. The recognition layer : F2 - Contains the long term memory of the system.
  3. Vigilance Parameter : ρ - A parameter that controls the generality of the memory. Larger ρ means more detailed memories, smaller ρ produces more general memories.

Types of ART :

Type	Remarks
ART 1	It is the simplest variety of ART networks, accepting only binary inputs.
ART 2	Extends network capabilities to support continuous inputs.
ART 3	ART 3 builds on ART-2 by simulating rudimentary neurotransmitter regulation of synaptic activity by incorporating simulated sodium (Na+) and calcium (Ca2+) ion concentrations into the system's equations, which results in a more physiologically realistic means of partially inhibiting categories that trigger mismatch resets.
Fuzzy ART	Fuzzy ART implements fuzzy logic into ART's pattern recognition, thus enhancing generalizability
ARTMAP	It is also known as Predictive ART, combines two slightly modified ART-1 or ART-2 units into a supervised learning structure where the first unit takes the input data and the second unit takes the correct output data, then used to make the minimum possible adjustment of the vigilance parameter in the first unit in order to make the correct classification.
Fuzzy ARTMAP	Fuzzy ARTMAP is merely ARTMAP using fuzzy ART units, resulting in a corresponding increase in efficiency.

Q.30 List the advantages of adaptive resonance theory.

Ans. : • It exhibits stability and is not disturbed by a wide variety of inputs provided to its network.

- It can be integrated and used with various other techniques to give more good results.
- It can be used for various fields such as mobile robot control, face recognition, land cover classification, target recognition, medical diagnosis, signature verification, clustering web users, etc.
- It has got advantages over competitive learning. The competitive learning lacks the capability to add new clusters when deemed necessary.
- It does not guarantee stability in forming clusters.

2.6 : Special Networks

Q.31 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.32 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.33 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.34 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.

Fill in the Blanks for Mid Term Exam

Q.1 The SOM algorithm is a _____ quantization algorithm, which provides a good approximation to the input space.

Q.2 _____ networks are a type of artificial neural network constructed from spatially localized kernel functions.

Q.3 An output neuron that wins the competition is called a _____ neuron.

Q.4 Hamming net uses _____ as a subnet to find the unit with the largest net input.

Q.5 In a _____ map, the neurons are placed at the nodes of a lattice that is usually one or two dimensional.

Q.6 A specific competitive net that performs Winner Take All (WTA) competition is the _____ .

Q.7 The _____ algorithm is based on unsupervised, competitive learning.

Q.8 Counter propagation network is a _____ winner-take-all competitive learning network.

Q.9 The weight vector for an output unit is often referred to as a _____ vector for the class tat the unit represents.

Q.10 _____ nets are designed to allow the user to control the degree of similarity of patterns placed on the same cluster.

Q.11 The ability of a net to respond to learn a new pattern equally well at any stage of learning is called _____.

Q.12 Counter-propagation networks are _____ networks based on a combination of input, clustering and output layers.

Q.13 In perceptron convergence theorem, the adaption weight vector is done by considering which appropriate parameter ?

Q.14 _____ is a specific technique for implemented in improving the efficiency of multi-layer feed forward network.

Q.15 _____ is most popular technique in which we start with a large multilayer perceptron with an adequate performance for the problem and then remove it by eliminating certain synaptic weights in an orderly fashion.

Q.16 A HAP model which comes under auto correlated associative memory is abbreviated as _____.

Multiple Choice Questions for Mid Term Exam

Q.1 Kohonen network trained in an _____ mode.

a supervised b unsupervised
c semi-supervised d All of these

Q.2 What is an activation value ?

a weighted sum of inputs
b threshold value
c main input to neuron
d none of the mentioned

Q.3 What is Hebb's rule of learning ?

a the system learns from its past mistakes
b the system recalls previous reference inputs and respective ideal outputs
c the strength of neural connection get modified accordingly
d none of the mentioned

Q.4 Why can't we design a perfect neural network ?

a full operation is still not known of biological neurons
b number of neuron is itself not precisely known
c number of interconnection is very large and is very complex
d all of these

Q.5 What was the main point of difference between the Adaline and perceptron model ?

a weights are compared with output

b sensory units result is compared with output

c analog activation value is compared with output

d all of the mentioned

Q.6 Heteroassociative memory can be an example of which type of network ?

a group of instars

b group of oustar

c either group of instars or outstars

d both group of instars or outstars

Q.7 The ability to learn how to do tasks on the data given for training or initial experience is know as _____.

a self-organization b adaptive learning

c fault tolerance d robustness

Q.8 The sigmoid activation function is generally identified in _____ shape.

a S b Z

c A d U

Q.9 In general the "error based learning algorithm" comes under _____.

a reinforced b supervised

c unsupervised d private

Answer Key for Fill in the Blanks

Q.1	vector	**Q.2**	Radial basis function
Q.3	winner-takes all	**Q.4**	Maxnet
Q.5	self-organizing	**Q.6**	Maxnet
Q.7	SOM	**Q.8**	unsupervised
Q.9	reference	**Q.10**	Adaptive resonance theory
Q.11	plasticity	**Q.12**	multilayer
Q.13	linearly separable	**Q.14**	back - propagation
Q.15	supervised	**Q.16**	Hopfield

Answer Key for Multiple Choice Questions

Q.1	b	**Q.2**	a	**Q.3**	c	**Q.4**	a
Q.5	c	**Q.6**	c	**Q.7**	b	**Q.8**	a
Q.9	b						

END...

UNIT - III

3 Deep Learning

3.1 : Introduction to Deep Learning

Q.1 Define deep learning.

Ans. : Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans : learn by example. Deep learning uses layers of algorithms to process data, understand human speech, and visually recognize objects.

Q.2 Explain deep learning. What are the challenges in deep learning ?

Ans. : • Deep Learning is a new area of machine learning research, which has been introduced with the objective of moving machine learning closer to one of its original goals.

- Deep learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound and text.
- 'Deep learning' means using a neural network with several layers of nodes between input and output. It is generally better than other methods on image, speech and certain other types of data because the series of layers between input and output do feature identification and processing in a series of stages, just as our brains seem to.
- Deep learning emphasizes the network architecture of today's most successful machine learning approaches. These methods are based on "deep" multi-layer neural networks with many hidden layers.

Challenges in Deep learning :

- They need to find and process massive datasets for training
- One of the reasons deep learning works so well is the large number of interconnected neurons, or free parameters, that allow for capturing subtle nuances and variations in data.
- Due to the sheer number of layers, nodes, and connections, it is difficult to understand how deep learning networks arrive at insights.
- Deep-learning networks are highly susceptible to the butterfly effect-small variations in the input data can lead to drastically different results, making them inherently unstable.

Q.3 List the application of deep learning.

Ans. : Applications :

1. Colorization of black and white images.
2. Adding sounds to silent movies.
3. Automatic machine translation.
4. Object classification in photographs.
5. Automatic handwriting generation.
6. Character text generation.
7. Image caption generation.
8. Automatic game playing.

Q.4 What is vanishing gradient problem ?

Ans. : • Vanishing gradient problem occurs when we try to train a neural network model using gradient based optimization techniques.

- Now when we do back-propagation i.e. moving backward in the network and calculating gradients of loss (error) with respect to the weights, the gradients tends to get smaller and smaller as we keep on moving backward in the network.
- This means that the neurons in the earlier layers learn very slowly as compared to the neurons in the later layers in the hierarchy. The earlier layers in the network are slowest to train.
- Gradient based methods learn a parameter's value by understanding how a small change in the parameter's value will affect the network's output.

- If a change in the parameter's value causes very small change in the network's output - the network just can't learn the parameter effectively, which is a problem.
- This is exactly what's happening in the vanishing gradient problem, the gradients of the network's output with respect to the parameters in the early layers become extremely small.
- That's a fancy way of saying that even a large change in the value of parameters for the early layers doesn't have a big effect on the output. Let's try to understand when and why does this problem happen.
- Vanishing gradient problem depends on the choice of the activation function. Many common activation functions 'squash' their input into a very small output range in a very non-linear fashion.
- For example, sigmoid maps the real number line onto a "small" range of [0, 1].
- As a result, there are large regions of the input space which are mapped to an extremely small range.
- In these regions of the input space, even a large change in the input will produce a small change in the output - hence the gradient is small.

Q.5 Why is deep learning useful ?

Ans. : • Manually designed features are often over-specified, incomplete and take a long time to design and validate.

- Learned features are easy to adapt, fast to learn.
- Deep learning provides a very flexible, (almost?) universal, learnable framework for representing world, visual and linguistic information.
- Can learn both unsupervised and supervised.
- Effective end-to-end joint system learning.
- Utilize large amounts of training data.

Q.6 What's the difference between machine learning and deep learning ?

Ans. : • Deep learning is a specialized form of machine learning. A machine learning workflow starts with relevant features being manually extracted from images. The features are then used to create a model that categorizes the objects in the image.

- With a deep learning workflow, relevant features are automatically extracted from images. In addition, deep learning performs "end-to-end learning", where a network is given raw data and a task to perform, such as classification, and it learns how to do this automatically.

Q.7 Explain limitations of deep learning.

Ans. :

1. **Data dependency :** In general, deep learning algorithms require vast amounts of training data to perform their tasks accurately.
2. **Lack of generalization :** Deep-learning algorithms are good at performing focused tasks but poor at generalizing their knowledge.
3. **Algorithmic bias :** Deep-learning algorithms are as good as the data they're trained on.
4. **Explainability :** Neural networks develop their behavior in extremely complicated ways.

Q.8 What is convolutional neural network ? List its limitation.

Ans. : • In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery.

- CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics.
- Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex.
- Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field.
- It is fully-connected structure does not scale to large images. The explicit assumption that the inputs are images.

- It allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement. It vastly reduce the amount of parameters in the network.
- A CNN is a neural network with some convolutional layers. A convolutional layer has a number of filters that does convolutional operation.
- CNNs run a small window over the input image at both training and testing time, the weights of the network that looks through this window can learn from various features of the input data regardless of their absolute position within the input.

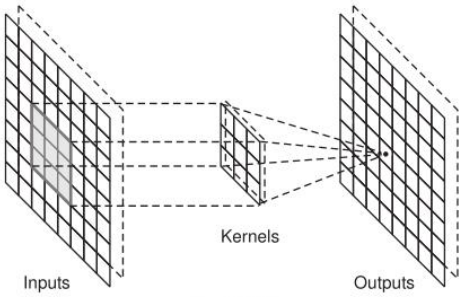


Fig. Q.8.1

- GoogLeNet was one of the first models that introduced the idea that CNN layers.
- A CNN compresses a fully connected network in three ways :
  1. Reducing number of connections.
  2. Shared weights on the edges.
  3. Max pooling further reduces the complexity.
- The CNN has 3 key properties :
  1. Locality : Allows more robustness against non-white noise where some bands are cleaner than the others.
  2. Weight sharing : Can also improve model robustness and reduce overfitting as each weight is learned from multiple frequency bands in the input instead of just from one single location.
  3. Pooling : The same feature values computed at different locations are pooled together and represented by one value.
- The limitations of the convolutional neural networks :
  1. Take fixed length vectors as input and produce fixed length vectors as output.
  2. Allow fixed amount of computational steps.

Q.9 Explain architecture of Convolution Neural Networks (ConvNet).

Ans. : • The architecture of a CNN is designed to take advantage of the 2D structure of an input image.

- CNN consists of a number of convolutional and sub-sampling layers optionally followed by fully connected layers.
- The input to a convolutional layer is "**m × m × r**" image where m is the height and width of the image and r is the number of channels.
- Fig. Q.9.1 shows architecture of convolution neural network.
- CNN architecture is made up of several layers that implement feature extraction, and then classification.
- The image is divided into receptive fields that feed into a convolutional layer, which then extracts features from the input image.
- The next step is pooling, which reduces the dimensionality of the extracted features (through down-sampling) while retaining the most important information (typically through max pooling).

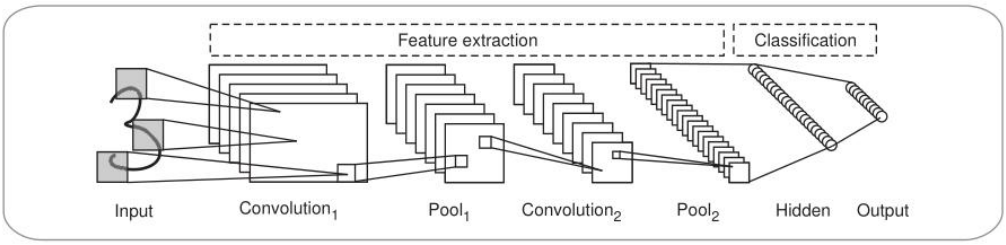


Fig. Q.9.1 Architecture of convolution neural network

- Another convolution and pooling step is then performed that feeds into a fully connected multilayer perceptron. The final output layer of this network is a set of nodes that identify features of the image. You train the network by using back-propagation.
- An input image is passed to the first convolutional layer. The convoluted output is obtained as an activation map. The filters applied in the convolution layer extract relevant features from the input image to pass further.
- Each filter shall give a different feature to aid the correct class prediction. In case we need to retain the size of the image, we use same padding (zero padding), otherwise valid padding is used since it helps to reduce the number of features.
- Pooling layers are then added to further reduce the number of parameters.
- Several convolution and pooling layers are added before the prediction is made. Convolutional layer help in extracting features.
- As we go deeper in the network more specific features are extracted as compared to a shallow network where the features extracted are more generic.
- The output layer in a CNN as mentioned previously is a fully connected layer, where the input from the other layers is flattened and sent so as the transform the output into the number of classes as desired by the network.
- The output is then generated through the output layer and is compared to the output layer for error generation.
- A loss function is defined in the fully connected output layer to compute the mean square loss. The gradient of error is then calculated.
- The error is then backpropagated to update the filter (weights) and bias values. One training cycle is completed in a single forward and backward pass.

Q.10 Write short note on Pooling layer.

Ans. : • Pooling layers were developed to reduce the number of parameters needed to describe layers deeper in the network.

- Pooling also reduces the number of computations required for training the network, or for simply running it forward during a classification task.
- Fig. Q.10.1 shows pooling layer.

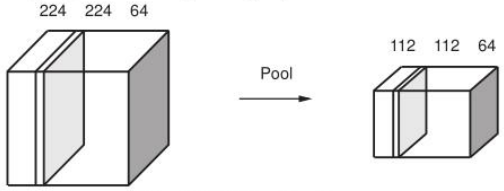


Fig. Q.10.1 Pooling layer

- Pooling layers provide some limited amount of translational and rotational invariance. However, it will not account for large translational or rotational perturbations, such as a face flipped by 180.
- Perturbations like these could only be detected if images of faces, rotated by 180, were present in the original training set.
- Accounting for all possible orientations of an object would require more filters, and therefore more weight parameters would be required.

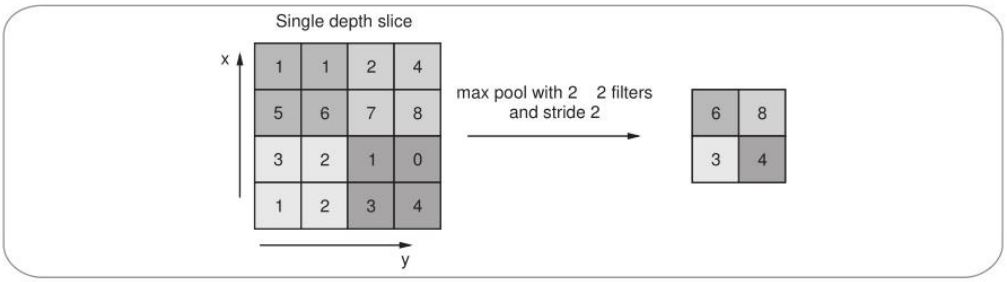
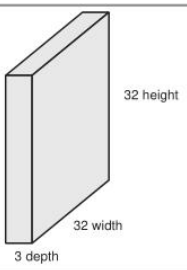
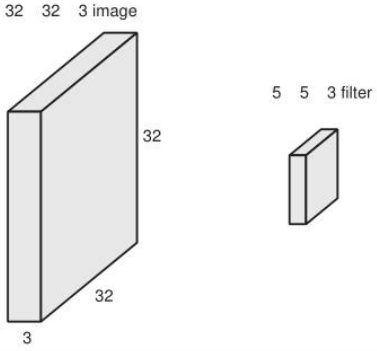
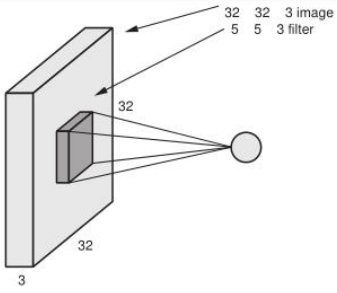


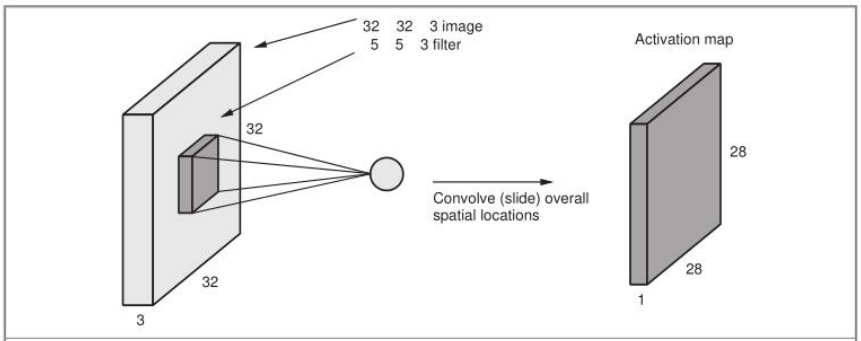
Fig. Q.10.2 Max pooling

- Its function is to progressively reduce the spatial size of the representation to reduce the amount of parameters and computation in the network. Pooling layer operates on each feature map independently.
- The most common approach used in pooling is max pooling.
- The pooling function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. Max pooling reports the maximum output within a rectangular neighborhood. Average pooling reports the average output.
- Pooling helps make the representation approximately invariant to small input translations.

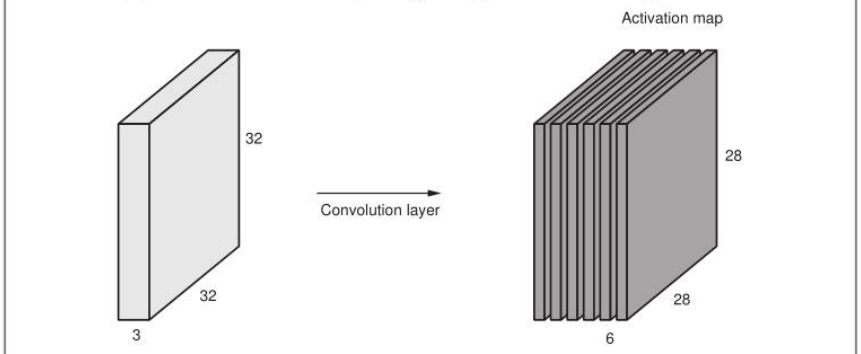
Q.11 Write short note on convolutional layer.

Ans. : The Conv layer is the core building block of a convolutional network that does most of the computational heavy lifting.

$32 \times 32 \times 3$ image is preserve spatial structure.	
Convolve the filter with the image i.e. "slide over the image spatially, computing dot products". Filters always extend the full depth of the input volume.	
1 number : The result of taking a dot product between the filter and a small $5 \times 5 \times 3$ chunk of the image.	



• For example, if we had 6 5 × 5 filters, we'll get 6 separate activation maps.



We stack these up to get a "new image" of size 28x28x6.

Q.12 Explain applications of convolutional neural network.

Ans. : • **Computer vision :** CNN are employed to identify the hierarchy or conceptual structure of an image. Instead of feeding each image into the neural network as one grid of numbers, the image is broken down into overlapping image tiles that are each fed into a small neural network.

- Speech recognition : CNN have been used recently in speech recognition and has given better results over deep neural networks. Robustness of CNN is enhanced when pooling is done at a local frequency region and over-fitting is avoided by using fewer parameters to extract low-level features.
- CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.
- They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.
- Video analysis : Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space.

- Drug discovery : CNNs have been used in drug discovery. Predicting the interaction between molecules and biological proteins can identify potential treatments.
- Natural language processing : CNNs have also explored natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks.

3.2 : Deep Feedforward Networks

Q.13 Why models are called feedforward ?

Ans. : Models are called feedforward because information flows through the function being evaluated from x, through the intermediate computations used to define f, and finally to the output y. There are no feedback connections in which outputs of the model are fed back into itself.

Q.14 Why feedforward neural network are called network ?

Ans. : Feedforward neural networks are called networks because they are typically represented by composing together many different functions. The model is associated with a directed acyclic graph describing how the functions are composed together.

Q.15 What is recurrent neural networks ?

Ans. : When feedforward neural networks are extended to include feedback connections, they are called recurrent neural networks.

Q.16 Write short note on Deep feedforward networks.

Ans. : • Deep feedforward networks is also called feedforward neural networks or multilayer perceptrons.

- Feedforward neural networks are called networks because they are typically represented by composing together many different functions. The model is associated with a directed acyclic graph describing how the functions are composed together.
- For example : consider 3 functions $f^{(1)}$, $f^{(2)}$, and $f^{(3)}$ connected in a chain, to form $f(x) = (f^{(1)} f^{(3)}(f^{(2)}(x)))$.
- These chain structures are the most commonly used structures of neural networks. In this case, $f^{(1)}$ is called the **first layer** of the network, $f^{(2)}$ is called the **second layer**, and so on.
- The overall length of the chain gives the depth of the model. It is from this terminology that the name "deep learning" arises.
- The final layer of a feedforward network is called the **output layer**. During neural network training, we drive f(x) to match $f^*(x)$.
- The training data provides us with noisy, approximate examples of $f^*(x)$ evaluated at different training points.
- The training examples specify directly what the output layer must do at each point x; it must produce a value that is close to y. The behavior of the other layers is not directly specified by the training data.
- The learning algorithm must decide how to use those layers to produce the desired output, but the training data does not say what each individual layer should do.
- Instead, the learning algorithm must decide how to use these layers to best implement an approximation of f^*.
- Because the training data does not show the desired output for each of these layers, these layers are called **hidden layers**.
- Finally, these networks are called *neural* because they are loosely inspired by neuroscience. Each hidden layer of the network is typically vector-valued. The dimensionality of these hidden layers determines the **width** of the model.

Q.17 What is use of hidden layer ?

Ans. : Feedforward networks have introduced the concept of a hidden layer, and this requires us to choose the activation functions that will be used to compute the hidden layer values.

Q.18 Explain learning concept of XOR.

Ans. : • XOR problem is a pattern recognition problem in neural network.

- The XOR function is an operation on two binary values, x1 and x2. When exactly one of these binary values is equal to 1, the XOR function returns 1. Otherwise, it returns 0.
- The XOR function provides the target function $y = f^*(x)$ that we want to learn. Our model provides a function $y = f(x; v)$ and our learning algorithm will adapt the parameters v to make f as similar as possible to f^*.
- Neural networks can be used to classify Boolean functions depending on their desired outputs.
- The XOR problem is not **linearly separable**. We cannot use a single layer perceptron to construct a straight line to partition the two dimensional input space into two regions, each containing only data points of the same class.

Q.19 List the types of activation function.

Ans. : Types of activation functions are : Sigmoid or Logistic, Tanh and ReLu.

Q.20 Define activation function. Explain the purpose of activation function in multilayer neural networks. Give any two activation functions.

Ans. : • An activation function f performs a mathematical operation on the signal output. The activation functions are chosen depending upon the type of problem to be solved by the network. There are a number of common activation functions in use with neural networks.

- Fig. Q.20.1 shows position of activation function. Unit step function is one of the activation function.

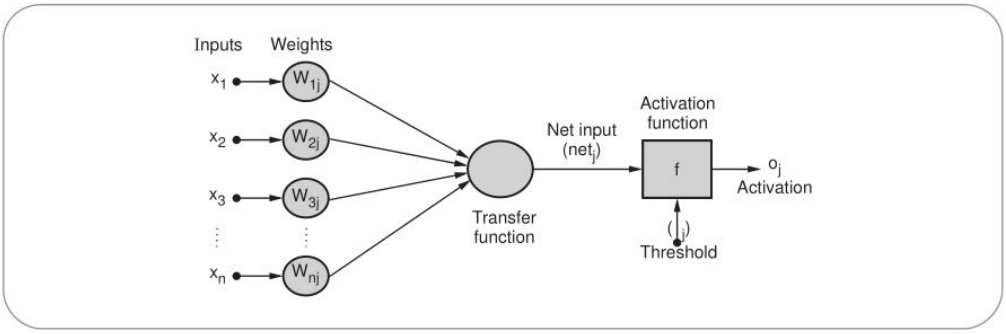


Fig. Q.20.1 Position of activation function

- The cell body itself is considered to have two functions. The first function is integration of all weighted stimuli symbolized by the summation sign. The second function is the activation which transforms the sum of weighted stimuli to an output value which is sent out through connection y.
- Typically the same activation function is used for all neurons in any particular layer. In a multi-layer network if the neurons have linear activation functions the capabilities are no better than a single layer network with a linear activation function. Hence in most cases nonlinear activation functions are used.

1. **Linear activation function :** The linear activation function will only produce positive numbers over the entire real number range. The linear activation function value is 0 if the argument is less than a lower boundary, increasing linearly from 0 to +1 for arguments equal or larger than the lower boundary and less than an upper boundary, and +1 for all arguments equal or greater than a given upper boundary.

2. **Sigmoid activation function :** The sigmoid function will only produce positive numbers between 0 and 1. The sigmoid activation function is most used for training data that is also between 0 and 1. It is one of the most used activation functions. A sigmoid function produces a curve with an "S" shape. Logistic and hyperbolic tangent functions are commonly used sigmoid functions. The sigmoid functions are extensively used in back propagation neural networks because it reduces the burden of complication involved during training phase.

$$\text{sig}(t) = \frac{1}{1+e^{-t}}$$

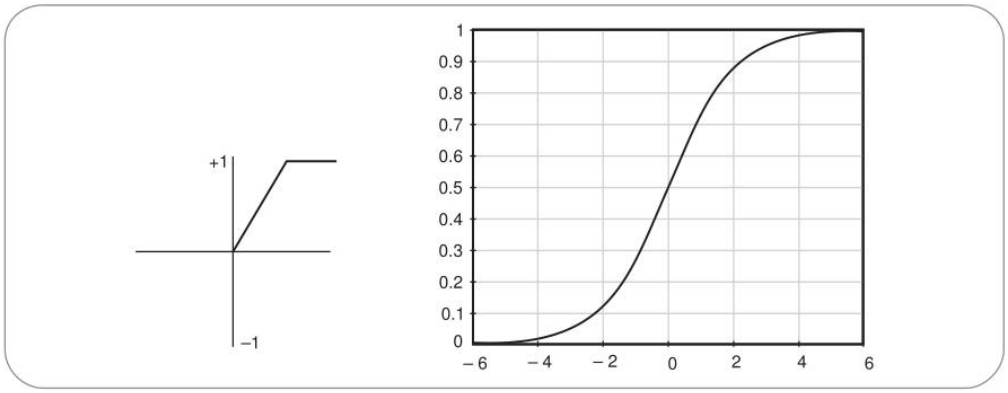


Fig. Q.20.2 (a) Linear functions **Fig. Q.20.2 (b) Sigmoid functions**

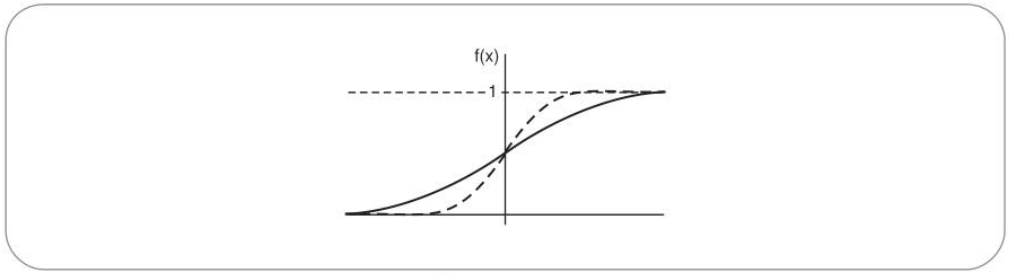


Fig. Q.20.2 (c) Binary sigmoid function

3. **Binary sigmoid :** The logistic function, which is a sigmoid function between 0 and 1 are used in neural network as activation function where the output values are either binary or varies from 0 to 1. It is also called as binary sigmoid or logistic sigmoid.

$$f(x) = \frac{1}{1+e^{-x}}$$

4. **Bipolar sigmoid :** A logistic sigmoid function can be scaled to have any range of values which may be appropriate for a problem. The most common range is from –1 to 1. This id called bipolar sigmoid.

$$f(x) = -1 + \frac{2}{1+e^{-x}}$$

The bipolar sigmoid is also closely related to the hyperbolic tangent function.

$$\tan h(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)} = \frac{1 - \exp(-2x)}{1 + \exp(-2x)}$$

Q.21 Write short note on Tanh and ReLU neurons.

Ans. : • Tanh is also like logistic sigmoid but better. The range of the tanh function is from (-1 to 1). Tanh is also sigmoidal (s - shaped).

- Fig. Q.21.1 shows tanh v/s Logistic Sigmoid.
- Tanh neuron is simply a scaled sigmoid neuron.
- Problems resolved by Tanh
  1. The output is not zero centered
  2. Small gradient of sigmoid function
- ReLU (Rectified Linear Unit) is the most used activation function in the world right now. Since, it is used in almost all the convolution neural networks or deep learning.
- Fig. Q.21.2 shows ReLU v/s Logistic Sigmoid.

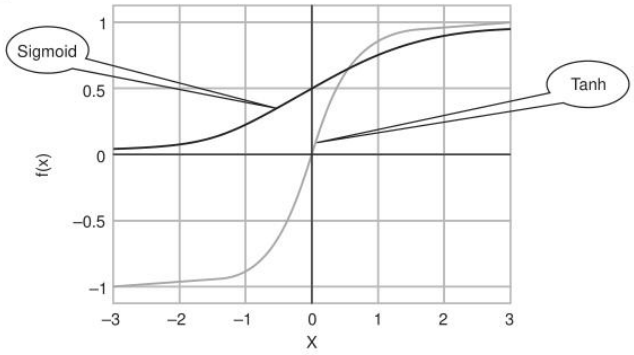


Fig. Q.21.1 : tanh v/s Logistic Sigmoid

- As you can see, the ReLU is half rectified (from bottom). f(z) is zero when z is less than zero and f(z) is equal to z when z is above or equal to zero.

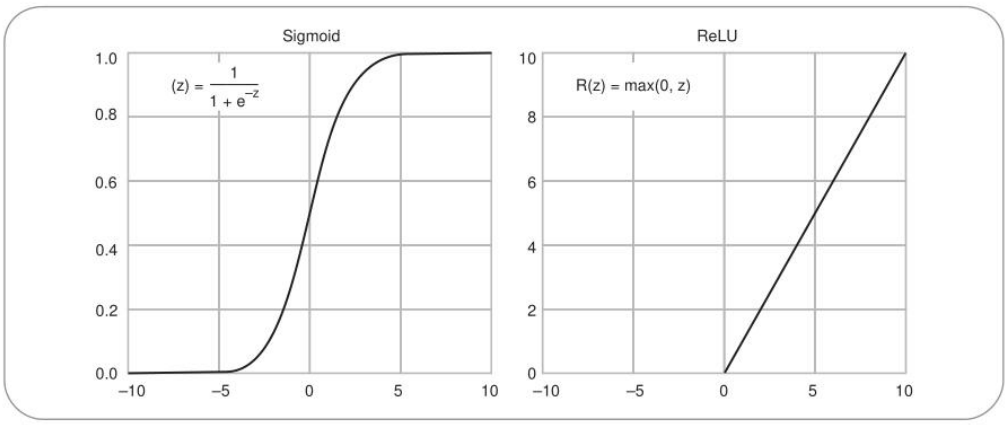


Fig. Q.21.2 : ReLU v/s Logistic Sigmoid

- Compared to tanh/sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLU can be implemented by simply thresholding a matrix of activations at zero.

Function	Advantages	Disadvantages
Sigmoid	1. Outputin range (0,1)	1. Saturated Neurons 2. Not zero entered 3. Small gradient 4. Vanishing gradient
Tanh	1. Zero centered, 2. Output in range(-1,1)	1. Saturated Neurons
ReLU	1. Computational efficiency, 2. Accelerated convergence	1. Dead Neurons, 2. Not zero centered

Q.22 List the pros and cons of ReLU.

Ans. : Pros :

1. Faster to compute
2. It helps to reduce vanishing gradient problem.
3. Sparse activation i.e. more robust to noise
4. Efficient to optimize, converges much faster than sigmoid or tanh.

Cons :

1. Very negative neuron cannot recover due to zero gradient.
2. Non zero centered output.

3.3 : Gradient-Based Learning

Q.23 What is gradient descent ?

Ans. : 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.

Q.24 What is cost function ?

Ans. : Cost function define as :

$$C(w, b) = \frac{1}{2N} \sum_{i=1}^{N} ||a(x_i) - y_i||^2$$

where x are the input vectors and y are there corresponding labels, both are determined. So the cost function chagnes it's value depending on the weights w and biases b. It is quadratic cost function.

Q.25 List the components of hidden units.

Ans. : Hidden units are composed of input vectors (x), computing an affine transformation (z) and element-wise nonlinear function.

Q.26 What is sigmoid unit ?

Ans. : A sigmoid unit is used when predicting the value of a binary variable. That means it can predict for only two cases. That also means that the only probability distribution which the sigmoid unit is able to predict is the Bernoulli distribution.

Q.27 What is use of softmax unit ?

Ans. : Softmax functions are most often used as the output of a classifier, to represent the probability distribution over n different classes. More rarely, softmax functions can be used inside the model itself, if we wish the model to choose between one of n different options for some internal variable.

Q.28 What is difference between linear unit and rectified linear unit ?

Ans. : The only difference between a linear unit and a rectified linear unit is that a rectified linear unit outputs zero across half its domain. This makes the derivatives through a rectified linear unit remain large whenever the unit is active. The gradients are not only large but also consistent.

Q.29 Write short note on softmax unit.

Ans. : • The softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function. But it also divides each output such that the total sum of the outputs is equal to 1.

- The output of the softmax function is equivalent to a categorical probability distribution, it tells you the probability that any of the classes are true.
  - Mathematically the softmax function is shown below, where z is a vector of the inputs to the output layer. And again, j indexes the output units, so j = 1, 2, ..., K.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

  - Each training image is labeled with the true digit and the goal of the network is to predict the correct label. So, if the input is an image of the digit 4, the output unit corresponding to 4 would be activated, and so on for the rest of the units.
  - The softmax function highlights the largest values and suppress other values.
  - The softmax can be used for any number of classes. It's also used for hundreds and thousands of classes, for example in object recognition problems where there are hundreds of different possible objects.

Q.30 Explain gradient descent algorithm. List the limitation of gradient descent.

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 $-\nabla f(x)$.
- 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.30.1 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 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.

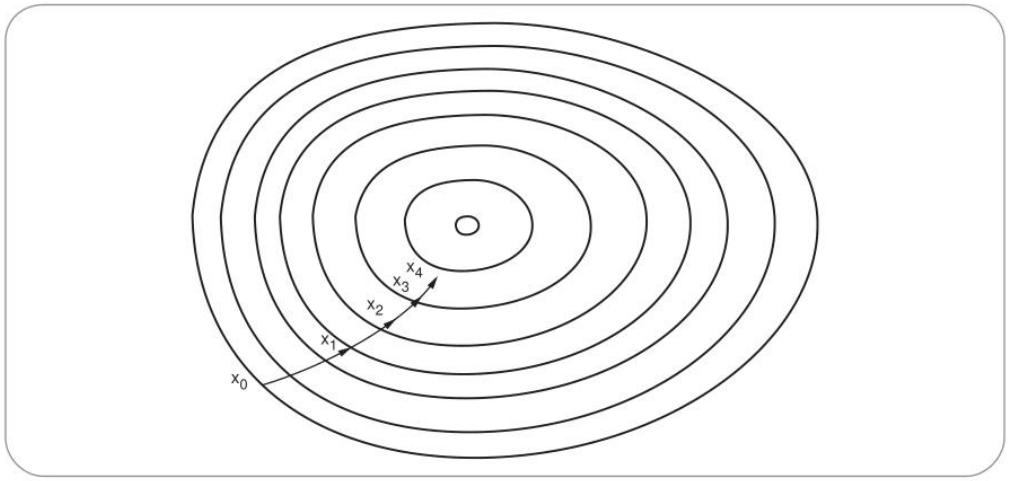


Fig. Q.30.1 Steepest descent method

Q.31 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.

3.4 : Architecture Design

Q.32 What is universal approximation theorem ?

Ans. : A single hidden layer neural network with a linear output unit can approximate any continuous function arbitrary well, given enough hidden units.

Q.33 What do you mean architecture design ?

Ans. : • Architecture refers to the overall structure of the network.

- Most neural networks are organized into groups of units called layers. These layers are arranged in a chain structure, with each layer being a function of the layer that preceded it.
- In this structure, the first layer is given by

$$h^{(1)} = g^{(1)}(W^{(1)T}x + b^{(1)})$$

- The second layer is given by

$$h^{(2)} = g^{(2)}(W^{(2)T}h^{(1)} + b^{(2)})$$

3.5 : Back-Propagation and Other Differentiation Algorithms

Q.34 What is forward propagation ?

Ans. : When we use a feedforward neural network to accept an input x and produce an output $\hat{y}$, information flows forward through the network. The inputs x provides the initial information that then propagates up to the hidden units at each layer and finally produces $\hat{y}$. This is called **forward propagation**.

Q.35 What is backpropagation ?

Ans. : • Backpropagation allows information to flow backwards from cost to compute the gradient.

- It is an algorithm for supervised learning of artificial neural networks using gradient descent.

- Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights.

Q.36 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.36.1 shows the backpropagation MLP for node j. The net input and output of node j is as follows :

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

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

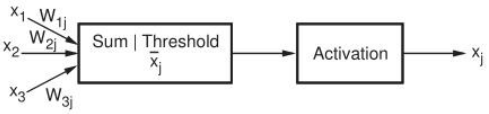


Fig. Q.36.1 Backpropagation MLP for node j

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 backpropagation MLP.
- Fig. Q.36.2 shows two layer backpropagation MLP.

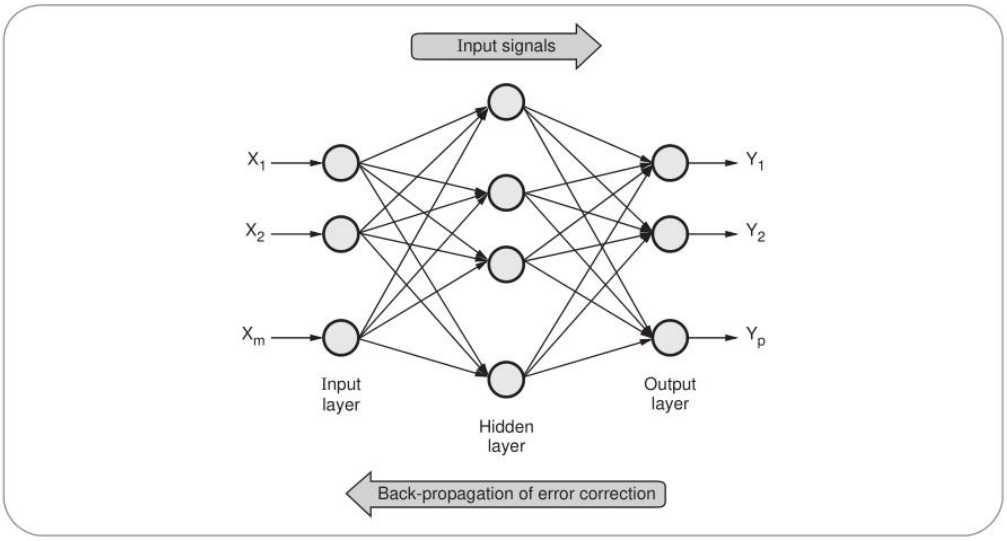


Fig. Q.36.2

- 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.37 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.38 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.39 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.

Q.40 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.41 Why backpropagation is also called as generalized delta rule ? ☞[JNTU : May-17, Marks-5]

Ans. : • The generalized delta rule is a mathematically derived formula used to determine how to update a neural network during a (back propagation) training step.

- A neural network learns a function that maps an input to an output based on given example pairs of inputs and outputs.
- A set number of input and output pairs are presented repeatedly, in random order during the training.
- The generalized delta rule is used repeatedly during training to modify weights between node connections.
- Before training, the network has connection weights initialized with small, random numbers. The purpose of the weight modifications is to reduce the overall network error, which means to reduce the difference between the actual and expected output.
- The backpropagation algorithm trains a given feed-forward multilayer neural network for a given set of input patterns with known classifications.
- When each entry of the sample set is presented to the network, the network examines its output response to the sample input pattern.
- The output response is then compared to the known and desired output and the error value is calculated. Based on the error, the connection weights are adjusted.
- The backpropagation algorithm is based on Widrow-Hoff delta learning rule in which the weight adjustment is done through mean square error of the output response to the sample input.
- The set of these sample patterns are repeatedly presented to the network until the error value is minimized.

Fill in the Blanks for Mid Term Exam

Q.1 _____ feedforward networks, also often called feedforward neural networks, or multilayer perceptron's.

Q.2 When feedforward neural networks are extended to include feedback connections, they are called _____ neural networks.

Q.3 A single-layer with one hidden unit also called _____.

Q.4 Gradient descent is an iterative _____ algorithm.

Q.5 ReLU is one of the most popular function which is used as _____ activation function in deep neural network.

Q.6 Training data does not show the desired output for each of these layers, these layers are called _____.

Q.7 _____ gradient descent applied to non-convex loss functions has no such convergence guarantee, and is sensitive to the values of the initial parameters.

Q.8 The iterative gradient-based optimization algorithms used to train _____.

Q.9 Rectified linear units are an excellent default choice of ____ unit.

Q.10 The _____ algorithm looks for the minimum of the error function in weight space using the method of gradient descent.

Q.11 Algebraic expressions and computational graphs both operate on ____, or variables that do not have specific values.

Q.12 The first order derivatives are called the gradient and the second order derivatives are called the _____ .

Multiple Choice Questions for Mid Term Exam

Q.1 ReLU stands for _____.

- a Rectified Link Unit
- b Rectified Large Unit
- c Rectified Layer Unit
- d Rectified Linear Unit

Q.2 The iterative gradient-based optimization algorithms used to train _____.

- a feedforward networks
- b backpropagation
- c activation function
- d hidden layer

Q.3 Hidden units are composed of _____, computing an affine transformation and element-wise nonlinear function.

- a output vector
- b input vector
- c activation function
- d all of these

Q.4 The backpropagation algorithm is used to find a local minimum of the _____.

- a neural network
- b activation function
- c error function
- d none of these

Q.5 The ____ function highlights the largest values and suppress other values.

- a Error
- b activation
- c backpropagation
- d softmax

Q.6 In back propagation algorithms a _____ determines the size of the weight adjustments made at each iteration and influences the rate of convergence.

- a ∞
- b μ
- c η
- d α

Answer Key for Fill in the Blanks

Q.1	Deep	**Q.2**	recurrent
Q.3	perceptron	**Q.4**	optimization
Q.5	hidden layer	**Q.6**	hidden layers
Q.7	Stochastic	**Q.8**	feedforward networks
Q.9	hidden	**Q.10**	backpropagation
Q.11	symbols	**Q.12**	Hessian matrix

Answer Key for Multiple Choice Questions

Q.1	d	**Q.2**	a	**Q.3**	b	**Q.4**	c
Q.5	b	**Q.6**	c				

END...

UNIT - IV

4 Regularization for Deep Learning

4.1 : Parameter Norm Penalties

Q.1 What is regularization ?

Ans. : Regularization refers to a set of different techniques that lower the complexity of a neural network model during training, and thus prevent the overfitting.

Q.2 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.2.1 shows overfiting.

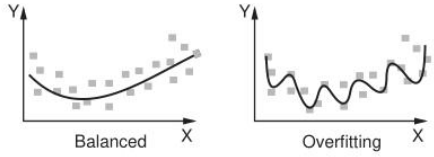


Fig. Q.2.1 : Overfitting

- 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.3 Explain underfitting. What are the reason for underfitting ? How do we know if we are underfitting or overfitting ?

Ans. : • Underfitting : If we put too few variables in the model, leaving out variables that could help explain the response, we are **underfitting**.

- Consequences :
  1. Fitted model is not good for prediction of new data - prediction is biased.
  2. Regression coefficients are biased.
  3. Estimate of error variance is too large.
- Underfitting examples :
  1. The learning time may be prohibitively large, and the learning stage was prematurely terminated.

2. The learner did not use a sufficient number of iterations.
3. The learner tries to fit a straight line to a training set whose examples exhibit a quadratic nature.

- To prevent under-fitting we need to make sure that :
  1. The network has enough hidden units to represent the required mappings.
  2. The network is trained for long enough that the error/cost function is sufficiently minimized.
- How do we know if we are underfitting or overfitting ?
  1. If by increasing capacity we decrease generalization error, then we are underfitting, otherwise we are overfitting.
  2. If the error in representing the training set is relatively large and the generalization error is large, then underfitting;
  3. If the error in representing the training set is relatively small and the generalization error is large, then overfitting;
  4. There are many features but relatively small training set.

Q.4 Explain L^2 parameter regularization.

Ans. :

- It is the hyperparameter whose value is optimized for better results. L^2 regularization is also known as weight decay as it forces the weights to decay towards zero.
- The weights are

$$\Omega(w) = \frac{1}{2}||w||_2^2 = \frac{1}{2}w^T w$$

- The update rule of gradient decent using L^2 norm penalty is

$$w \leftarrow (1-\in\alpha)w-\in\nabla_w j(w)$$

- The weights multiplicatively shrink by a constant factor at each step.
- L^2 regularization drives the weights closer to origin by adding a regularization term.
- The L^2 regularization has the intuitive interpretation of heavily penalizing peaky weight vectors and preferring diffuse weight vectors.
- Due to multiplicative interactions between weights and inputs this has the appealing property of encouraging the network to use all of its inputs a little rather that some of its inputs a lot.
- The L^2 regularization causes the learning algorithm to "perceive" the input X as having higher variance, which makes it shrink the weights on features whose covariance with the output target is low compared to this added variance.
- In L^2 regularization, regularization term is the sum of square of all feature weights. It forces the weights to be small but does not make them zero and does non sparse solution.

Q.5 What is L^1 norm parameter regulation ?

Ans. :

- A regression model that uses L^1 regularization technique is called Lasso Regression.
- Lasso Regression adds "absolute value of magnitude" of coefficient as penalty term to the loss function.
- Formally, L^1 regularization on the model parameter w is defined as

$$\Omega(\theta) = ||\omega||_1 = \sum|\omega_1|$$

that is, as the sum of absolute values of the individual parameters.

- In L^1, we have:

$$\text{Cost function} = \text{Loss} + \frac{\lambda}{2m} * \sum||w||$$

- In this, we penalize the absolute value of the weights. Unlike L^2, the weights may be reduced to zero.
- Hence, it is very useful when we are trying to compress our model. Otherwise, we usually prefer L^2 over it.
- In L^1 regularization, the weights shrink by a constant amount toward 0. In L^2 regularization, the weights shrink by an amount which is proportional to w.
- And so when a particular weight has a large magnitude, $|w|$, L^1 regularization shrinks the weight much less than L^2 regularization does.
- By contrast, when $|w|$ is small, L^1 regularization shrinks the weight much more than L^2 regularization. The net result is that L^1 regularization tends to concentrate the weight of

the network in a relatively small number of high-importance connections, while the other weights are driven toward zero.

Q.6 What is difference between L^1 and L^2 regulation ?

Ans. :

L^1 regulation	L^2 regulation
Calculate the sum of the absolute values of the weights, called L^1.	Calculate the sum of the squared values of the weights, called L^2.
L^1 has a sparse solution.	L^2 has a non sparse solution.
It has built in feature selection.	There is no feature selection.
L^1 is robust to outliers	L^2 is not robust to outliers
L^1 generates model that are simple and interpretable but cannot learn complex patterns.	L^2 regularization is able to learn complex data patterns.
L^1 has multiple solutions.	L^2 has one solution.
It is equivalent to MAP Bayesian estimation with Laplace prior.	It is equivalent to MAP Bayesian estimation with Gaussian prior.
A regression model that uses L^1 regularization technique is called Lasso Regression.	model which uses L^2 is called Ridge Regression.

Q.7 Explain applications of convolutional neural network.

Ans. : • Computer vision : CNN are employed to identify the hierarchy or conceptual structure of an image. Instead of feeding each image into the neural network as one grid of numbers, the image is broken down into overlapping image tiles that are each fed into a small neural network.

- Speech recognition : CNN have been used recently in speech recognition and has given better results over deep neural networks. Robustness of CNN is enhanced when pooling is done at a local frequency region and over-fitting is avoided by using fewer parameters to extract low-level features
- CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.
- They have applications in image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.
- **Video analysis :** Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space.
- **Drug discovery :** CNNs have been used in drug discovery. Predicting the interaction between molecules and biological proteins can identify potential treatments.
- **Natural language processing :** CNNs have also explored natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks.

Q.8 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.8.1 shows overfiting.

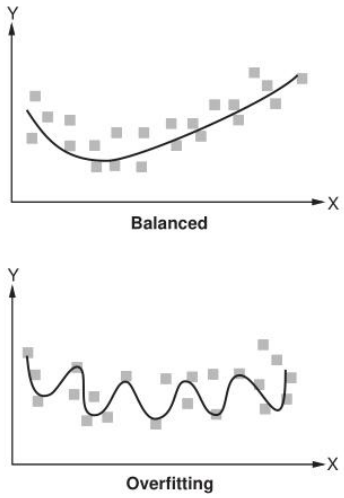


Fig. Q.8.1 Overfitting

- 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.

4.2 : Norm Penalties as Constrained Optimization

Q.9 What is goal of penalty functions ?

Ans. :

- The goal of penalty functions is to convert constrained problems into unconstrained problems by introducing an artificial penalty for violating the constraint.
- In the exact penalty functions methods, the original constrained optimization problem is replaced by an unconstrained problem, in which the objective function is the sum of a certain "merit" function and a penalty term which reflects the constraint set.
- The penalty term is obtained by multiplying a suitable function, which represents the constraints, by a positive parameter c, called the penalty parameter.
- A given penalty-parameter c is called an exact penalty-parameter when every solution of the given extremum problem can be found by solving the unconstrained optimization problem with the penalty function associated with c.

Q.10 Explain norm penalties as constrained optimization.

Ans. :

- If Ω is the L^2 norm, then the weights are constrained to lie in an L^2 ball. If Ω is the L^1 norm, then the weights are constrained to lie in a region of limited L^1 norm.
- Usually we do not know the size of the constraint region that we impose by using weight decay with coefficient α^* because the value of α^* does not directly tell us the value of k.
- In principle, one can solve for k, but the relationship between k and α^* depends on the form of J.
- While we do not know the exact size of the constraint region, we can control it roughly by increasing or decreasing α in order to grow or shrink the constraint region.
- Larger α will result in a smaller constraint region. Smaller α will result in a larger constraint region.

Q.11 What are the reasons to use explicit constraints rather than penalties ?

Ans. :

- Reason to use explicit constraints and reprojection rather than enforcing constraints with penalties is that penalties can cause nonconvex optimization procedures to get stuck in local minima corresponding to small θ.
- When training neural networks, this usually manifests as neural networks that train with several "dead units." These are units that do not contribute much to the behavior of the function learned by the network because the weights going into or out of them are all very small.
- When training with a penalty on the norm of the weights, these configurations can be locally optimal, even if it is possible to significantly reduce J by making the weights large.

4.3 : Dataset Augmentation and Noise Robustness

Q.12 What is dataset augmentation ?

Ans. :

- Data augmentation is the process of increasing the amount and diversity of data. We do not collect new data, rather we transform the already present data.
- Data augmentation is an integral process in deep learning, as in deep learning we need large amounts of data and in some cases it is not feasible to collect thousands or millions of images, so data augmentation comes to the rescue.
- It helps us to increase the size of the dataset and introduce variability in the dataset.
- Data augmentation involves the process of creating new data points by manipulating the original data. For example, for images, this can be done by rotating, resizing, cropping, and more.
- This process increases the diversity of the data available for training models in deep learning without having to actually collect new data. This then, generally speaking, improves the performance of deep learning models.
- Random Erasing is a data augmentation method used for training convolutional neural networks that involves randomly erasing a rectangular region in an image. Images with occlusions are then generated. This makes a model robust to occlusion and reduces the chances of overfitting.
- Injecting noise in the input to a neural network can also be seen as a form of data augmentation. For many classi?cation and even some regression tasks, the task should still be possible to solve even if small random noise is added to the input.
- One way to improve the robustness of neural networks is simply to train them with random noise applied to their inputs. Input noise injection is part of some unsupervised learning algorithms, such as the denoising autoencoder.
- Noise injection also works when the noise is applied to the hidden units, which can be seen as doing dataset augmentation at multiple levels of abstraction.

Q.13 What is dropout ? How it solve problem of overfitting ?

Ans. : • Dropout was a technique developed for preventing overfitting of a network to the particular variations of the training set.

- Deep neural networks with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks.
- Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem.
- The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much.
- During training, dropout samples from an exponential number of different "thinned" networks.
- At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single un-thinned network that has smaller weights.

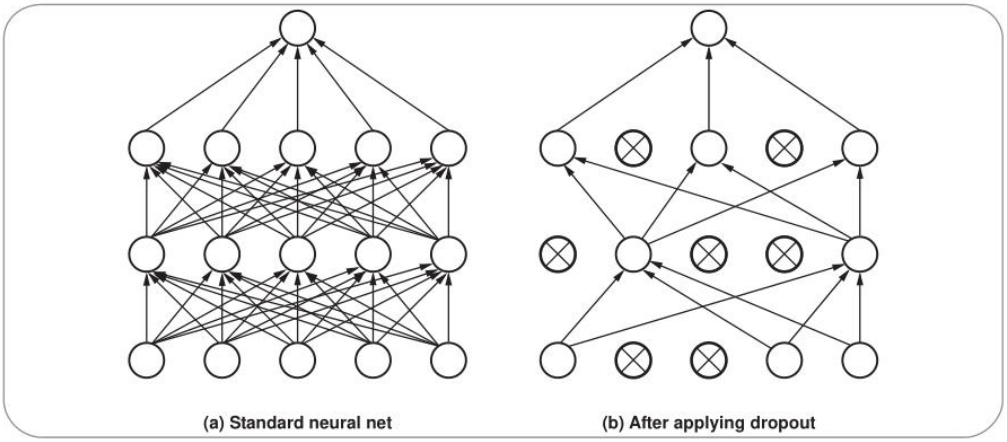


Fig. Q.13.1

- Fig. Q.13.1 shows neural network and dropout neural network.
- The term "dropout" refers to dropping out units (hidden and visible) in a neural network.
- With limited training data, however, many of these complicated relationships will be the result of sampling noise, so they will exist in the training set but not in real test data even if it is drawn from the same distribution.
- This leads to many overfitting methods have been developed for reducing it. These include stopping the training as soon as performance on a validation set starts to get worse, introducing weight penalties of various kinds such as L1 and L2 regularization and soft weight sharing.

4.4 : Multi-task learning and Early Stopping

Q.14 What is multi-task learning ?

Ans. :

- Multi-Task learning is a sub-field of Machine Learning that aims to solve multiple different tasks at the same time, by taking advantage of the similarities between different tasks.
- Multitask learning is a way to improve generalization by pooling the examples arising out of several tasks. Fig Q.14.1 shows multi-tasking.

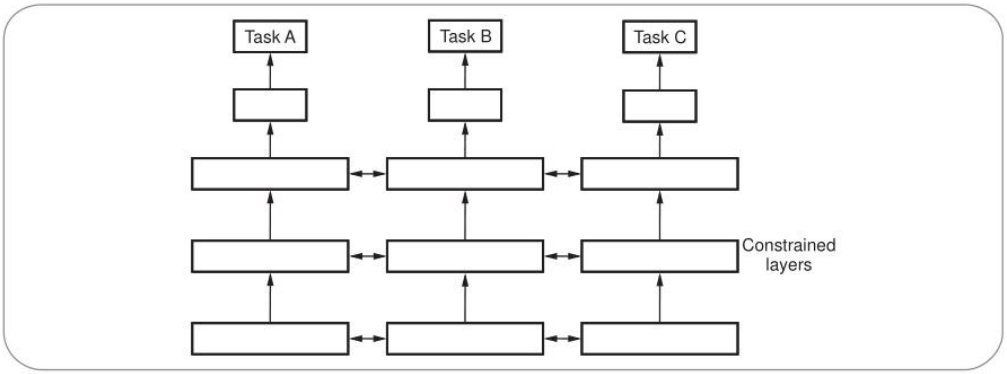


Fig. Q.14.1

- Different supervised task shared the same input and some intermediate-level representation.
- We can view multi-task learning as a form of inductive transfer. Inductive transfer can help improve a model by introducing an inductive bias, which causes a model to prefer some hypotheses over others.
- For instance, a common form of inductive bias is ?1 regularization, which leads to a preference for sparse solutions.
- Multi-tasking learning are of two types: hard or soft parameter sharing of hidden layers.
- Hard parameter: It is generally applied by sharing the hidden layers between all tasks, while keeping several task-specific output layers. Hard parameter sharing greatly reduces the risk of overfitting.
- In soft parameter sharing on the other hand, each task has its own model with its own parameters. The distance between the parameters of the model is then regularized in order to encourage the parameters to be similar.

Q.15 Discuss about early stopping.

Ans. :

- Early stopping is a technique that is very often used when training neural networks, as well as with some other iterative machine learning algorithms.
- Early stopping is a technique for controlling overfitting in machine learning models, especially neural networks, by stopping training before the weights have converged.
- This means we can obtain a model with better validation set error by returning to the parameter setting at the point in time with the lowest validation set error.
- Every time the error on the validation set improves, we store a copy of the model parameters.
- When the training algorithm terminates, we return these parameters, rather than the latest parameters.
- The algorithm terminates when no parameters have improved over the best recorded validation error for some pre-specified number of iterations. This strategy is known as early stopping.
- It is probably the most commonly used form of regularization in deep learning.
- Early stopping is an unobtrusive form of regularization, in that it requires almost no change in the underlying training procedure, the objective function, or the set of allowable parameter values.
- Early stopping may be used either alone or in conjunction with other regularization strategies.
- Early stopping requires a validation set, which means some training data is not fed to the model
- Early stopping is also useful because it reduces the computational cost of the training procedure.

4.5 : Parameter Typing and Parameter Sharing

Q.16 What is a parametric machine learning algorithm and nonparametric machine learning algorithm ?

Ans. : Parametric machine learning algorithm :

- Parametric model is one that can be parameterized by a finite number of parameters.
- Learning model that summarizes data with a set of parameters of fixed size is called a parametric model.
- No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs.
- Given the parameters, future predictions (x) are independent of the observed data (D) : $\mathbf{P(x|\theta, D) = P(x|\theta)}$ therefore θ capture everything there is to know about the data.
- So the complexity of the model is bounded even if the amount of data is unbounded. This makes them not very flexible.
- Some more examples of parametric machine learning algorithms include : Logistic regression, linear discriminant analysis, perceptron, naive bayes and simple neural networks.
- Benefits of Parametric Machine Learning Algorithms :
  1. **Simpler** : These methods are easier to understand and interpret results.

2. **Speed :** Parametric models are very fast to learn from data.
3. **Less data :** They do not require as much training data and can work well even if the fit to the data is not perfect.

- Limitations of parametric machine learning algorithms :
  1. **Constrained :** By choosing a functional form these methods are highly constrained to the specified form.
  2. **Limited complexity :** The methods are more suited to simpler problems.
  3. **Poor Fit :** In practice the methods are unlikely to match the underlying mapping function.

Nonparametric model :

- A nonparametric model is one which cannot be parametrized by a fixed number of parameters.
- Non-parametric models assume that the data distribution cannot be defined in terms of such a finite set of parameters. But they can often be defined by assuming an infinite dimensional θ. Usually we think of θ as a function.
- The amount of information that θ can capture about the data D can grow as the amount of data grows. This makes them more flexible.
- Nonparametric methods seek to best fit the training data in constructing the mapping function, whilst maintaining some ability to generalize to unseen data. As such, they are able to fit a large number of functional forms.
- An easy to understand nonparametric model is the k-nearest neighbors algorithm that makes predictions based on the k most similar training patterns for a new data instance.
- The method does not assume anything about the form of the mapping function other than patterns that are close are likely have a similar output variable.
- Examples of nonparametric machine learning algorithms are :
  1. k-Nearest Neighbors
  2. Decision Trees like CART and C4.5
  3. Support Vector Machines

- Benefits of nonparametric machine learning algorithms :
  1. **Flexibility :** Capable of fitting a large number of functional forms.
  2. **Power :** No assumptions (or weak assumptions) about the underlying function.
  3. **Performance :** Can result in higher performance models for prediction.
- Limitations of nonparametric machine learning algorithms :
  1. **More data :** Require a lot more training data to estimate the mapping function.
  2. **Slower :** A lot slower to train as they often have far more parameters to train.
  3. **Overfitting :** More of a risk to overfit the training data and it is harder to explain why specific predictions are made.

Q.17 Difference between parametric and non-parametric modeling.

Ans. :

Parametric modeling	Non-parametric modeling
Parametric : Data are drawn from a probability distribution of specific form up to unknown parameters. It uses single global model.	Nonparametric : Data are drawn from a certain unspecified probability distribution. There is no single global model; local models are estimated as they are needed.
Flexible discovery.	Easy idea.
Example : Logistic regression, linear discriminant analysis, perceptron, naive bayes and simple neural networks.	Example : k-nearest neighbors, decision trees like cart and support vector machines.
Benefit : Requires less data, simpler and speed.	Benefit : Flexibility, power and performance.
Limitation : Poor fit and limited complexity	Limitation : More data required, slower and overfitting.
It requires less data.	It requires more data.

Q.18 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.19 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.20 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.21 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.

4.6 : Bagging and Other Ensemble Methods

Q.22 Explain ensemble learning method.

Ans. : • The idea of ensemble learning is to employ multiple learners and combine their predictions. If we have a committee of M models with uncorrelated errors, simply by averaging them the average error of a model can be reduced by a factor of M.

- Unfortunately, the key assumption that the errors due to the individual models are uncorrelated is unrealistic; in practice, the errors are typically highly correlated, so the reduction in overall error is generally small.
- Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications.
- Ensemble of classifiers is a set of classifiers whose individual decisions combined in some way to classify new examples.
- Ensemble methods combine several decision trees classifiers to produce better predictive performance than a single decision tree classifier. The main principle behind the ensemble model is that a group of weak learners come together to form a strong learner, thus increasing the accuracy of the model.
- There are two approaches for combining models : **voting and stacking.**
- In **voting**, no learning takes place at the meta level when combining classifiers by a voting scheme. Label that is most often assigned to a particular instance is chosen as the correct prediction when using voting.
- **Stacking** is concerned with combining multiple classifiers generated by different learning algorithms $L_1, \ldots, L_N$ on a single dataset S, which is composed by a feature vector $S_i = (x_i, t_i)$.

- The stacking process can be broken into two phases :
  1. Generate a set of base-level classifiers $C_1, \ldots, C_N$ Where $C_i = L_i(S)$
  2. Train a meta-level classifier to combine the outputs of the base-level classifiers
- Fig. Q.22.1 shows stacking frame.
- The training set for the meta-level classifier is generated through a leave-one-out cross validation process.

$$\forall i = 1, \ldots n \text{ and } \forall k = 1, \ldots\ldots, N : C_k^i$$
$$= L_k(S - s_i)$$

- The learned classifiers are then used to generate predictions for $s_i : \hat{y}_i^k = C_k^i(x_i)$
- The meta-level dataset consists of examples of the form $((\hat{y}_i^1, \ldots, \hat{y}_i^n), y_i)$, where the features are the predictions of the base-level classifiers and the class is the correct class of the example in hand.
- Why do ensemble methods work?
- Based on one of two basic observations :
  1. Variance reduction : If the training sets are completely independent, it will always helps to average an ensemble because this will reduce variance without affecting bias (e.g.,bagging) and reduce sensitivity to individual data points.
  2. Bias reduction : For simple models, average of models has much greater capacity than single model Averaging models can reduce bias substantially by increasing capacity, and control variance by Citting one component at a time.

Q.23 What is bagging ? Explain bagging steps. List its advantages and disadvantages.

Ans. : • Bagging is also called Bootstrap aggregating. Bagging and boosting are meta-algorithms that pool decisions from multiple classifiers. It creates ensembles by repeatedly randomly resampling the training data.

- Bagging was the first effective method of ensemble learning and is one of the simplest methods of arching. The meta-algorithm, which is a special case of the model averaging, was originally designed for classification and is usually applied to decision tree models, but it can be used with any type of model for classification or regression.

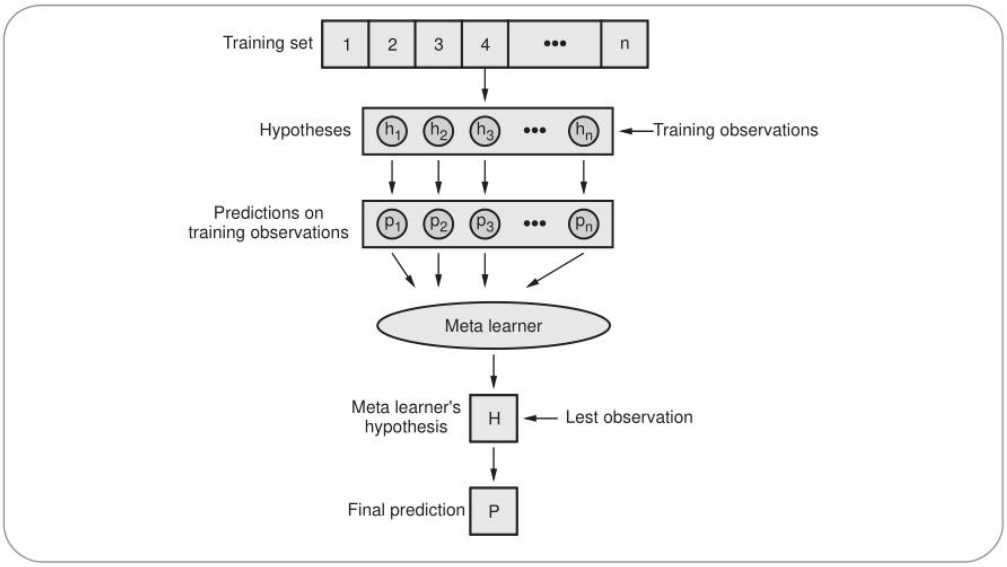


Fig. Q.22.1 Stacking frame

- Ensemble classifiers such as bagging, boosting and model averaging are known to have improved accuracy and robustness over a single model. Although unsupervised models, such as clustering, do not directly generate label prediction for each individual, they provide useful constraints for the joint prediction of a set of related objects.
- For given a training set of size n, create m samples of size n by drawing n examples from the original data, with replacement. Each *bootstrap sample* will on average contain 63.2 % of the unique training examples, the rest are replicates. It combines the m resulting models using simple majority vote.
- In particular, on each round, the base learner is trained on what is often called a "bootstrap replicate" of the original training set. Suppose the training set consists of n examples. Then a bootstrap replicate is a new training set that also consists of n examples, and which is formed by repeatedly selecting uniformly at random and with replacement n examples from the original training set. This means that the same example may appear multiple times in the bootstrap replicate, or it may appear not at all.
- It also decreases error by decreasing the variance in the results due to *unstable learners,* algorithms (like decision trees) whose output can change dramatically when the training data is slightly changed.
- **Pseudocode :**

1. Given training data $(x_1, y_1), \ldots, (x_m, y_m)$
2. For t = 1,..., T :
   a. Form bootstrap replicate dataset S_t by selecting m random examples from the training set with replacement.
   b. Let h_t be the result of training base learning algorithm on S_t.
3. Output combined classifier :
   $H(x) = \text{majority } (h_1(x), \ldots, h_T(x))$.

Bagging Steps :

1. Suppose there are N observations and M features in training data set. A sample from training data set is taken randomly with replacement.
2. A subset of M features is selected randomly and whichever feature gives the best split is used to split the node iteratively.
3. The tree is grown to the largest.
4. Above steps are repeated n times and prediction is given based on the aggregation of predictions from n number of trees.

Advantages of Bagging :

1. Reduces over-fitting of the model.
2. Handles higher dimensionality data very well.
3. Maintains accuracy for missing data.

Disadvantages of Bagging :

1. Since final prediction is based on the mean predictions from subset trees, it won't give precise values for the classification and regression model.

Q.24 Explain boosting steps. List advantages and disadvantages of boosting.

Ans. : Boosting Steps :

1. Draw a random subset of training samples d_1 without replacement from the training set D to train a weak learner C_1
2. Draw second random training subset d_2 without replacement from thetraining set and add 50 percent of the samples that were previously falsely classified/misclassified to train a weak learner C_1
3. Find the training samples d_3 in the training set D on which C_1 and C_1 disagree to train a third weak learner C_3
4. Combine all the weak learners via majority voting.

Advantages of Boosting :

1. Supports different loss function.
2. Works well with interactions.

Disadvantages of Boosting :

1. Prone to over-fitting.
2. Requires careful tuning of different hyper-parameters.

4.7 : Adversarial Training

Q.25 What is adversarial training ? Explain.

Ans. :

- Adversarial machine learning is a technique employed in the field of machine learning which attempts to fool models through malicious input. This technique can be applied for a variety of reasons, the most common being to attack or cause a malfunction in standard machine learning models.
- Adversarial training helps to illustrate the power of using a large function family in combination with aggressive regularization.
- Purely linear models, like logistic regression, are not able to resist adversarial examples because they are forced to be linear.
- Neural networks are able to represent functions that can range from nearly linear to nearly locally constant and thus have the flexibility to capture linear trends in the training data while still learning to resist local perturbation.
- Adversarial examples also provide a means of accomplishing semi-supervised learning. Adversarial examples generated using not the true label but a label provided by a trained model are called virtual adversarial examples.

4.8 : Tangent Distance

Q.26 Write short note on tangent propagation.

Ans. :

- The tangent prop algorithm trains a neural net classifier with an extra penalty to make each output f(x) of the neural net locally invariant to known factors of variation.
- As with the tangent distance algorithm, the tangent vectors are derived a priori, usually from the formal knowledge of the effect of transformations, such as translation, rotation, and scaling in images.
- The tangent distance algorithm is a popular method that estimates the manifold distance by employing a linear approximation of the transformation manifolds.
- Tangent prop has been used for supervised learning and reinforcement learning.
- Tangent propagation is closely related to dataset augmentation.
- Tangent propagation is also related to double backprop and adversarial training.
- Double backprop regularizes the Jacobian to be small, while adversarial training finds inputs near the original inputs and trains the model to produce the same output on these as on the original inputs.
- Tangent propagation and dataset augmentation using manually specified transformations both require that the model be invariant to certain specified directions of change in the input.
- The manifold tangent classifier eliminates the need to know the tangent vectors a priori.

Fill in the Blanks for Mid Term Exam

Q.1 In machine learning, _____ is way to prevent over-fitting.

Q.2 Regularization reduces over-fitting by adding a penalty to the _____ function.

Q.3 _____ simplifies a machine learning problem by choosing which subset of the available features should be used.

Q.4 Underfitting occurs when the model is not able to obtain a suficiently low error value on the _____.

Q.5 Overfitting occurs when the gap between the training error and _____ is too large.

Q.6 The sparsity property induced by _____ regularization has been used extensively as a feature selection mechanism.

Q.7 Each penalty is a product between a coefficient, called a _____ multiplier, and a function representing whether the constraint is satisfied.

Q.8 Injecting noise in the input to a neural network can also be seen as a form of _____.

Q.9 _____ can be interpreted as a way of regularizing a neural network by adding noise to its hidden units

Q.10 Early stopping requires a _____, which means some training data is not fed to the model

Q.11 Bagging means _____

Q.12 The technique called _____ constructs an ensemble with higher capacity than the individual models.

Q.13 Adversarial examples generated using not the true label but a label provided by a trained model are called _____ examples.

Q.14 A related technique _____ Distance is used to build invariance properties into distance-based methods such as nearest-neighbor classifiers

Q.15 Tangent propagation is closely related to ____.

Multiple Choice Questions for Mid Term Exam

Q.1 L^2 regularization is also known as _____

- a Ridge regression
- b Lasso regression
- c Linear regression
- d All of these

Q.2 In L^2 regularization, regularization term is the _____ of all feature weights.

- a Absolute value
- b sum of square
- c square
- d sum

Q.3 Input noise injection is part of some _____ learning algorithms, such as the denoising autoencoder.

- a Semisupervised
- b supervised
- c unsupervide
- d Deep

Q.4 Calculate the sum of the absolute values of the weights, called _____ L^1.

a L^2 b L^1

c L^0 d none

Q.5 Dropout boosting trains the entire ensemble to jointly maximize the log-likelihood on the _____.

a training set

b testing set

c training set and testing set

d None

Answer Key for Fill in the Blanks

Q.1	regularization	**Q.2**	loss
Q.3	Feature selection	**Q.4**	training set
Q.5	test error	**Q.6**	L^1
Q.7	Karush-Kuhn-Tucker	**Q.8**	data augmentation
Q.9	Dropout	**Q.10**	validation set
Q.11	bootstrap aggregating	**Q.12**	boosting
Q.13	virtual adversarial	**Q.14**	Tangent
Q.15	dataset augmentation		

Answer Key for Multiple Choice Questions

Q.1	a	**Q.2**	b
Q.3	c	**Q.4**	b
Q.5	b		

END...

UNIT - V

5 Optimization for Train Deep Models

5.1 : Challenges in Neural Network Optimization

Q.1 What is convex optimization ?

Ans. : Convex optimization involves a function in which there is only one optimum, corresponding to the global optimum (maximum or minimum). There is no concept of local optima for convex optimization problems, making them relatively easy to solve.

Q.2 What is non-convex optimization ?

Ans : Non-convex optimization involves a function which has multiple optima, only one of which is the global optima. Depending on the loss surface, it can be very difficult to locate the global optima.

Q.3 What is empirical risk minimization ?

Ans : • Given a training set S and a function space H, empirical risk minimization is the class of algorithms that look at S and select f_S as,

$$f_S = \arg\min_{f \in H} I_S[f]$$

- The training process based on minimizing this average training error is known as empirical risk minimization.

Q.4 Discuss about maximum - likelihood estimation.

Ans : • Maximum likelihood, also called the maximum likelihood method, is the procedure of finding the value of one or more parameters for a given statistic which makes the known likelihood distribution a maximum.

- Maximum likelihood estimation is a totally analytic maximization procedure. It applies to every form of censored or multi-censored data, and it is even possible to use the technique across several stress cells and estimate acceleration model parameters at the same time as life distribution parameters.
- Moreover, MLEs and likelihood functions generally have very desirable large sample properties :

1. They become unbiased minimum variance estimators as the sample size increases
2. They have approximate normal distributions and approximate sample variances that can be calculated and used to generate confidence bounds
3. Likelihood functions can be used to test hypotheses about models and parameters

- The likelihood of the observed data, given the model parameters Θ , as the conditional probability that the model, M, with parameters Θ, produces *x[1] ,…, x[n].*

$$L(\Theta) = \Pr(x[1], \ldots, x[n] \mid \Theta, M),$$

- In MLE we seek the model parameters, Θ, that maximize the likelihood.

Likelihood Method

- Observations are outcomes of random experiments. The outcome is represented by a random variable (x). The distribution of possible outcomes is given by probability distribution.
- The same observations can be generated by different models and the different observations may be generated by the same model. The probability model predicts an outcome and associates a probability with each outcome.
- Suppose the entire result of an experiment is a collection of numbers (x) and suppose the joint pdf for the data x is a function that depends on a set of parameters θ.

$f(\bar{x} ; \bar{\theta})$

Then the likelihood function is given as :

$$L(\bar{\theta}) = f(\bar{x} ; \bar{\theta})$$

- Consider 'n' independent observations of $x : x_1, x_2, x_3, \ldots, x_{n-1}, x_n$ where x follows $f(x;\theta)$. The joint pdf for the whole data sample is as follows :

$$f(x_1, x_2, \ldots, x_n ;\theta) = \prod_{i=1}^{n} f(x_i ;\theta)$$

The likelihood function for above is as follows :

$$L(\bar{\theta}) = \prod_{i=1}^{n} f(x_i ;\bar{\theta})$$

- The maximum likelihood estimator is given by following formula :

$$d(X_1, X_2, X_3, \ldots, X_{n-1}, X_n) = \frac{\sum_{i=1}^{n} X_i}{n}$$

$$= \frac{1}{n} \sum_{i=1}^{n} X_i$$

- The sample standard deviation is given by

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

Q.5 Which factors are consider for minibatch sizes ?

Ans. : Following factors are consider :

1. Larger batches provide a more accurate estimate of the gradient, but with less than linear returns.
2. Multicore architectures are usually underutilized by extremely small batches.
3. If all examples in the batch are to be processed in parallel, then the amount of memory scales with the batch size. For many hardware setups this is the limiting factor in batch size.
4. Some kinds of hardware achieve better runtime with specific sizes of arrays.
5. Small batches can offer a regularizing effect, perhaps due to the noise they add to the learning process. Generalization error is often best for a batch size of 1.

Q.6 List the challenges in optimization.

Ans. : Challenges are ill-conditioning, local minima, plateaus, saddle points and other flat regions, cliffs and exploding gradients, long-term dependencies, inexact gradients and poor correspondence between local & global structure.

Q.7 Explain ill conditioning problem of neural network optimization.

Ans. : • The ill-conditioning problem is generally believed to be present in neural network training problems.

- Ill-conditioning can manifest by causing SGD to get "stuck" in the sense that even very small steps increase the cost function.
- Ill-conditioning of the Hessian matrix is a prominent problem in most numerical optimization problems, convex or otherwise.
- In multiple dimensions, there is a diferent second derivative for each direction at a single point.
- The condition number of the Hessian at this point measures how much the second derivatives differ from each other.
- When the Hessian has a large condition number, gradient descent performs poorly. This is because in one direction, the derivative increases rapidly, while in another direction, it increases slowly.
- Gradient descent is unaware of this change in the derivative so it does not know that it needs to explore preferentially in the direction where the derivative remains negative for longer.

Q.8 Define weight space symmetry.

Ans. : If we have m layers with n units each, then there are $(n!)^m$ ways of arranging the hidden units. This kind of non-identifiability is known as weight space symmetry.

Q.9 What is saddle point ?

Ans. : A saddle point is a point where the Hessian matrix has both positive and negative eigen values.

Q.10 What do you mean exploding gradients ?

Ans. : • Large updates to weights during training can cause a numerical overflow or underflow often referred to as "exploding gradients."

- A common and relatively easy solution to the exploding gradients problem is to change the derivative of the error before propagating it

backward through the network and using it to update the weights.

- Two approaches include rescaling the gradients given a chosen vector norm and clipping gradient values that exceed a preferred range. Together, these methods are referred to as "gradient clipping."

Q.11 What is stochastic gradient descent ?

Ans. : Much of machine learning can be written as an optimization problem.

- Example loss functions : Logistic regression, linear regression, principle component analysis, neural network loss.
- A very efficient way to train logistic models is with **Stochastic Gradient Descent (SGD)**.
- One challenge with training on power law data (i.e. most data) is that the terms in the gradient can have very different strengths.
- The idea behind stochastic gradient descent is iterating a weight update based on the gradient of loss function :

$$\overline{w}(k+1) = \overline{w}(k) - \gamma \nabla L(\overline{w})$$

- Logistic regression is designed as a **binary classifer** (output say {0, 1}) but actually **outputs the probability** that the input instance is in the "1" class.
- A logistic classifier has the form :

$$p(X) = \frac{1}{1+\exp(-X\beta)}$$

where $X = (X_1, \ldots, X_n)$ is a vector of features.

- Stochastic gradient has some serious limitations however, especially if the gradients vary widely in magnitude. Some coefficients change very fast, others very slowly.
- This happens for text, user activity and social media data (and other power-law data), because gradient magnitudes scale with feature frequency, i.e. over several orders of magnitude.
- It is not possible to set a single learning rate that trains the frequent and infrequent features at the same time.
- An example of stochastic gradient descent with perceptron loss is shown as follows :

from sklearn.linear_model import SGDClassifier

Q.12 What is momentum ?

Ans. : Momentum methods in the context of machine learning refer to a group of tricks and techniques designed to speed up convergence of first order optimization methods like gradient descent.

5.2 : Parameter Initialization Strategies and Algorithms with Adaptive Learning Rates

Q.13 Write short note on parameter initialization strategies.

Ans. : • Training algorithms for deep learning models are usually iterative and thus require the user to specify some initial point from which to begin the iterations.

- The initial point can determine whether the algorithm converges at all, with some initial points being so unstable that the algorithm encounters numerical difficulties and fails altogether.
- When learning does converge, the initial point can determine how quickly learning converges and whether it converges to a point with high or low cost.
- Points of comparable cost can have wildly varying generalization error, and the initial point can affect the generalization as well.
- If two hidden units with the same activation function are connected to the same inputs, then these units must have different initial parameters.
- If they have the same initial parameters, then a deterministic learning algorithm applied to a deterministic cost and model will constantly update both of these units in the same way.

Q.14 Explain the following :
a) AdaGrad b) Adam

Ans. : a) AdaGrad

- The AdaGrad algorithm is just a variant of preconditioned stochastic gradient descent.
- The AdaGrad individually adapts the learning rates of all model parameters by scaling them inversely proportional to the square root of the sum of all the historical squared values of the gradient.

- The parameters with the largest partial derivative of the loss have a correspondingly rapid decrease in their learning rate, while parameters with small partial derivatives have a relatively small decrease in their learning rate. The net effect is greater progress in the more gently sloped directions of parameter space.
- AdaGrad is designed to converge rapidly when applied to a convex function.
- One of Adagrad's main benefits is that it eliminates the need to manually tune the learning rate.
- Adagrad's main weakness is its accumulation of the squared gradients in the denominator : Since every added term is positive, the accumulated sum keeps growing during training. This in turn causes the learning rate to shrink and eventually become infinitesimally small, at which point the algorithm is no longer able to acquire additional knowledge.
- AdaGrad shrinks the learning rate according to the entire history of the squared gradient and may have made the learning rate too small before arriving at such a convex structure.

b) Adam

- Adaptive Moment Estimation (Adam) is another method that computes adaptive learning rates for each parameter. It keeps an exponentially decaying average of past gradients.
- Adam is generally regarded as being fairly robust to the choice of hyperparameters, though the learning rate sometimes needs to be changed from the suggested default.
- In Adam, momentum is incorporated directly as an estimate of the first-order moment (with exponential weighting) of the gradient.
- The weight update for Adam is given by :

Compute gradient : $g \leftarrow \frac{1}{m} \nabla_\theta \sum_i L(f(x^{(i)};\theta), y^{(i)})$

$$t \leftarrow t+1$$

Update biased first moment estimate :

$$s \leftarrow \rho_1 s + (1-\rho_1) g$$

Update biased second moment estimate :

$$r \leftarrow \rho_2 r + (1-\rho_2)\, g \odot g$$

Correct bias in first moment : $\hat{s} \leftarrow \frac{s}{1-\rho_1^t}$

Correct bias in second moment : $\hat{r} \leftarrow \frac{r}{1-\rho_2^t}$

Compute update : $\Delta\theta = -\epsilon \frac{\hat{s}}{\sqrt{\hat{r}} + \delta}$

(Operations applied element-wise)

Apply update : $\theta \leftarrow \theta + \Delta\theta$

- Since s and r are initialized as zeros, it is observed that, bias during the initial steps of training thereby adding a correction term for both the moments to account for their initialization near the origin.

5.3 : Approximate Second-Order Methods

Q.15 Explain newton method.

Ans. : • In contrast to first order gradient methods, second order methods make use of second derivatives to improve optimization.

- Most widely used second order method is Newton's method. It is described in more detail here emphasizing neural network training.
- It is based on Taylor's series expansion to approximate J(θ) near some point θ_0 ignoring derivatives of higher order.
- Taylor's series to approximate J(θ) near θ_0

$$J(\theta) \approx J(\theta_0) + (\theta-\theta_0)^T \nabla_\theta J(\theta_0) + \frac{1}{2}(\theta-\theta_0)^T H(\theta-\theta_0)$$

where H is the Hessian of J wrt θ evaluated at θ_0.

- Solving for the critical point of this function we obtain the Newton parameter update rule.

$$\theta^* = \theta_0 - H^{-1} \nabla_\theta J(\theta_0)$$

- Thus for a quadratic function (with positive definite H) by rescaling the gradient by H-1 Newton's method directly jumps to the minimum.
- If objective function is convex but not quadratic (there are higher-order terms) this update can be iterated yielding the training algorithm given next.

Q.16 What is conjugate gradient method ?

Ans. : The conjugate gradient method is the most prominent iterative method for solving sparse systems of linear equations. It is a method to effciently avoid the calculation of the inverse Hessian by iteratively descending conjugate directions.

5.4 : Optimization Strategies and Meta-Algorithms Applications

Q.17 What is batch normalization ?

Ans. : • Batch normalization is one of the most exciting innovations in Deep learning that has significantly stabilized the learning process and allowed faster convergence rates.

- Most of the Deep Learning networks are compositions of many layers or functions and the gradient with respect to one layer is taken considering the other layers to be constant. However, in practise all the layers are updated simultaneously and this can lead to unexpected results.
- For example, let $y^* = x\, W^1 W^2 \ldots W^{10}$. Here, y^* is a linear function of x but not a linear function of the weights. Suppose the gradient is given by g and we now intend to reduce y^* by 0.1
- Using first-order Taylor Series approximation, taking a step of $\in \mathbf{g}$ would reduce y^* by $\in \mathbf{g}'\mathbf{g}$.
- The updates to one layer is so strongly dependent on the other layers, choosing an appropriate learning rate is tough.
- Batch normalization takes care of this problem by using an efficient reparameterization of almost any deep network.
- Given a matrix of activations **(H)**, the normalization is given by : $\mathbf{H}' = (\mathbf{H} - \mu)/\sigma$, where the subtraction and division is broadcasted.

$$\mu = \frac{1}{m}\sum_i H_i$$

$$\sigma = \sqrt{\delta + \frac{1}{m}\sum_i (H-\mu)_i^2}$$

Q.18 What is coordinate descent ?

Ans. : In some cases, it may be possible to solve an optimization problem quickly by breaking it into separate pieces. If we minimize f(x) with respect to a single variable x_i, then minimize it with respect to another variable x_j, and so on, repeatedly cycling through all variables, we are guaranteed to arrive at a (local) minimum. This practice is known as coordinate descent.

Q.19 Explain how deep learning helps in speech recognition problem.

Ans. : • Speech recognition (is also known as Automatic Speech Recognition (ASR), or computer speech recognition) is the process of converting a speech signal to a sequence of words, by means of an algorithm implemented as a computer program.

- Automatic Speech Recognition (ASR) is the task of producing a text transcription of the audio signal from a speaker.
- Large Vocabulary Speech Recognition (LVSR) involves a large or open vocabulary and we will also take it to imply a speaker independent recognizer.
- The most general LVSR systems may also need to operate on spontaneous as well as read speech with audio data from an uncontrolled recording environment corrupted by both unstructured and structured noise.
- Traditional LVSR recognizers are based on Hidden Markov Models (HMMs) with Gaussian Mixture Model (GMM) emission distributions, n-gram language models, and use beam search for decoding.
- GMM-HMM acoustic models are trained with the Expectation Maximization (EM) algorithm using a procedure that trains progressively more sophisticated models using an initial alignment from the previous model.
- In speech recognition, the main goal of the feature extraction step is to compute a parsimonious sequence of feature vectors providing a compact representation of the given input signal.
- The feature extraction is usually performed in three stages.

1. The first stage is called the speech analysis or the acoustic front end. It performs some kind of spectro temporal analysis of the signal and generates raw features describing the envelope of the power spectrum of short speech intervals.
2. The second stage compiles an extended feature vector composed of static and dynamic features.
3. Finally, the last stage transforms these extended feature vectors into more compact and robust vectors that are then supplied to the recognizer.

- In speech recognition a supervised pattern classification system is trained with labeled examples; that is, each input pattern has a class label associated with it.
- Pattern classifiers can also be trained in an unsupervised fashion. For example in a technique known as vector quantization, some representation of the input data is clustered by finding implicit groupings in the data.
- The resulting table of cluster centers is known as a codebook, which can be used to index new vectors by finding the cluster center that is closest to the new vectors.
- Once a feature selection or classification procedure finds a proper representation, a classifier can be designed using a number of possible approaches.

Q.20 What is natural language processing ?

Ans. : • Natural language processing is a form of artificial intelligence that helps computers read and respond by simulating the human ability to understand everyday language.

- Many organizations use NLP techniques to optimize customer support, improve the efficiency of text analytics by easily finding the information they need, and enhance social media monitoring.
- For example, banks might implement NLP algorithms to optimize customer support; a large consumer products brand might combine natural language processing and semantic analysis to improve their knowledge management strategies and social media monitoring.

Fill in the Blanks for Mid Term Exam

Q.1 Optimization algorithms that use the entire _____ set are called batch or deterministic gradient methods.

Q.2 Optimization algorithms that use only a _____ example at a time are sometimes called stochastic and sometimes online methods.

Q.3 Empirical risk minimization is prone to _____ .

Q.4 A _____ loss function acts as a proxy to empirical risk while being "nice" enough to be optimized efficiently

Q.5 In contrast to standard optimization, training algorithms do not halt at _____ , but when early stopping halt criterion is satisfied.

Q.6 _____ Gradient Method is the most prominent iterative method for solving sparse systems of linear equations.

Multiple Choice Questions for Mid Term Exam

Q.1 Adam stands for _____ .

a Adaptive Moment Estimation
b Adaptive Deep Moment
c Adaptive Moment Learning
d None of these

Q.2 A saddle point is a point where the _____ has both positive and negative eigen values.

a diagonal matrix
b square matrix
c Hessian matrix
d scalar matrix

Q.3 Which of the following is challenges of optimization ?

a Ill-conditioning b Local minima
c Plateaus d All of thesee

Q.1 Polyak averaging consists of averaging several points in the trajectory through parameter space visited by an _____ algorithm.

a non-optimization b optimization
c both (a) and (b) d None

Answer Key for Fill in the Blanks

Q.1	training	Q.2	single
Q.3	overftting	Q.4	surrogate
Q.5	local minima	Q.6	Conjugate

Answer Key for Multiple Choice Questions

Q.1	a	Q.2	c	Q.3	d	Q.4	b

END...

SOLVED MODEL QUESTION PAPER

Neural Networks and Deep Learning (R18 Pattern)

Solved Paper
B.Tech., IV-II (CSE/IT)

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 (25 Marks)

Q.1 **a)** *What is artificial neural network ?* **(Refer Q.1 of Chapter - 1)** **[2]**

b) *Define auto associative memory.* **(Refer Q.32 of Chapter - 1)** **[3]**

c) *What is counter-propagation network ?* **(Refer Q.21 of Chapter - 2)** **[2]**

d) *What is winner-take-all learning network ?* **(Refer Q.5 of Chapter - 2)** **[3]**

e) *List the application of deep learning.* **(Refer Q.3 of Chapter - 3)** **[2]**

f) *What is forward propagation ?* **(Refer Q.34 of Chapter - 3)** **[3]**

g) *What is regularization ?* **(Refer Q.1 of Chapter - 4)** **[2]**

h) *What is difference between L^1 and L^2 regulation ?* **(Refer Q.6 Chapter - 4)** **[3]**

i) *What is convex optimization ?* **(Refer Q.1 of Chapter - 5)** **[2]**

j) *What is natural language processing ?* **(Refer Q.20 of Chapter - 5)** **[3]**

PART - B (50 Marks)

Q.2 **a)** *What is Bidirectional Associative Memory (BAM) ?* **(Refer Q.36 of Chapter - 1)** **[5]**

b) *What is an Adaline ? Explain in detail.* **(Refer Q.21 of Chapter - 1)** **[5]**

OR

Q.3 **a)** *Describe auto-associative memory.* **(Refer Q.33 of Chapter - 1)** **[5]**

b) *Explain useful properties and capabilities of neural network.* **(Refer Q.6 of Chapter - 1)** **[5]**

Q.4 **a)** *Explain hamming net with diagram.* **(Refer Q.9 of Chapter - 2)** **[5]**

b) *Draw and explain the architecture of ART. What is its use and types ?* **(Refer Q.29 of Chapter - 2)** **[5]**

OR

Q.5 **a)** *Write short note on learning vector quantization.* **(Refer Q.20 of Chapter - 2)** **[5]**

b) *Explain unsupervised learning.* **(Refer Q.2 of Chapter - 2)** **[5]**

Q.6 **a)** *Explain architecture of Convolution Neural Networks (ConvNet).* **(Refer Q.9 of Chapter - 3)** **[5]**

b) *Define activation function. Explain the purpose of activation function in multilayer neural networks. Give any two activation functions.* **(Refer Q.20 of Chapter - 3)** **[5]**

OR

Q.7 **a)** *What is vanishing gradient problem ?* **(Refer Q.4 of Chapter - 3)** **[5]**

b) *Explain gradient descent algorithm. List the limitation of gradient descent.* **(Refer Q.30 of Chapter - 3)** **[5]**

Q.8 **a)** *Explain overfitting. What are the reason for overfitting ?* **(Refer Q.2 of Chapter - 4)** **[5]**

b) *Explain applications of convolutional neural network.* **(Refer Q.7 of Chapter - 4)** **[5]**

OR

Q.9 **a)** *What is dropout ? How it solve problem of overfitting?* **(Refer Q.13 of Chapter - 4)** **[5]**

b) *What is a parametric machine learning algorithm and nonparametric machine learning algorithm ?* **(Refer Q.16 of Chapter - 4)** **[5]**

Q.10 **a)** *Explain how deep learning helps in speech recognition problem.* **(Refer Q.19 of Chapter - 5)** **[5]**

b) *What is batch normalization ?* **(Refer Q.17 of Chapter - 5)** **[5]**

OR

Q.11 **a)** *Which factors are consider for minibatch sizes ?* **(Refer Q.5 of Chapter - 5)** **[5]**

b) *What is stochastic gradient descent ?* **(Refer Q.11 of Chapter - 5)** **[5]**

END...