ANN Question Bank

1. Defne ANN and Neural computing.

ANN

An artificial neural network consists of a pool of simple processing units which communicate by sending signals to each other over a large number of weighted connections.

There are two basic reasons why we are interested in building artificial neural networks(ANNs):
Technical viewpoint: Some problems such as character recognition or the prediction of future states of a system require massively parallel and adaptive processing.
Biological viewpoint: ANNs can be used to replicate and simulate components of the human (or animal) brain ,thereby giving us insight into natural information processing.

An artificial neural network(ANN) is either a hardware implementation or a computer program which strives to simulate the information processing capabilities of its biological exemplar. ANNs are typically composed of a great number of interconnected artificial neurons. The artificial neurons are simplified models of their biological counterparts.

ANN is a technique for solving problems by constructing software that works like our brains.

Neural Computing

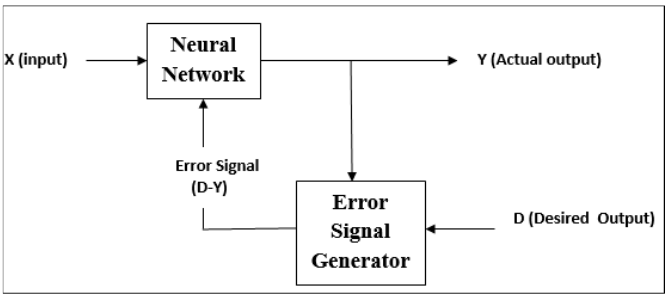
Neural computation is the information processing performed by networks of neurons. Artificial Neural Networks work in a way similar to that of their biological inspiration. They can be considered as **weighted directed graphs where the neurons could be compared to the nodes and the connection between two neurons as weighted edges**.

Artificial Neural Networks are made up of layers and layers of connected input units and output units called neurons. A single layer neural network is called a perceptron. Multiple hidden layers may also be present in an artificial neural network. The input units(receptor), connection weights, summing function, computation and output units (effectors) are what makes up an artificial neuron. The weight value of a connection is the strength of the specified connection between neurons. Weights are randomly initialized and adjusted via an optimization algorithm to map aggregations of input stimuli to a desired output function.

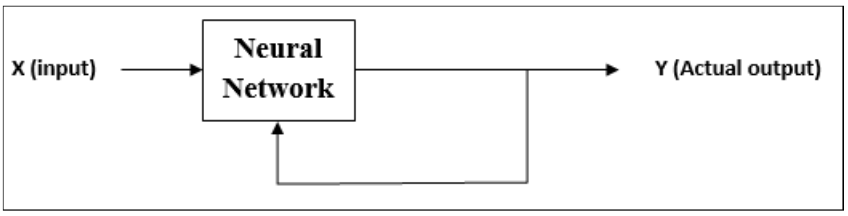
2. Distinguish between Supervised and Unsupervised Learning.

Supervised learning algorithms are trained using labeled data.	Unsupervised learning algorithms are trained using unlabeled data.
Supervised learning model takes direct feedback to check if it is predicting correct output or not.	Unsupervised learning model does not take any feedback.
Supervised learning model predicts the output.	Unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find the hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized in **Classification** and **Regression** problems.	Unsupervised Learning can be classified in **Clustering** and **Associations** problems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
Supervised learning model produces an accurate result.	Unsupervised learning model may give less accurate result as compared to supervised learning.
Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can	Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine

Supervised learning is not close to true Artificial intelligence as in this, we first train the model for each data, and then only it can predict the correct output.	Unsupervised learning is more close to the true Artificial Intelligence as it learns similarly as a child learns daily routine things by his experiences.
It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Clustering, KNN, and Apriori algorithm.



Supervised
Providing the network with a series of sample inputs and comparing the output with the expected responses.



Unsupervised
Most similar input vector is assigned to the same output unit

3. **Mention the characteristics of problems suitable for ANNs.**

- training data is noisy, complex sensor data
- also problems where symbolic algos are used (decision tree learning (DTL)) - ANN and DTL produce results of comparable accuracy
- instances are attribute-value pairs, attributes may be highly correlated or independent, values can be any real value
- target function may be discrete-valued, real-valued or a vector
- training examples may contain errors
- long training times are acceptable
- requires fast eval. of learned target func.
- humans do NOT need to understand the learned target func.

4. List some applications of ANN.

- Aerospace
- Automotive
- Banking
- Credit Card Activity Checking
- Defense
- Electronics
- Entertainment
- Financial
- Industrial
- Insurance
- Insurance
- Manufacturing
- Medical
- Oil and Gas
- Robotics
- Speech
- Securities
- Telecommunications
- Transportation

5. **What are the design parameters of ANN?**

- Determine the network properties:
  - Network topology
  - Types of connectivity
  - Order of connections
  - Weight range
- Determine the node properties:
  - Activation range
- Determine the system dynamics
  - Weight initialization scheme
  - Activation – calculating formula
  - Learning rule

6. **Explain the three classifications of ANNs based on their functions. Explain them in brief.**

- **Feedback ANN –** In these type of ANN, the output goes back into the network to achieve the best-evolved results internally. The feedback network feeds information back into itself and is well suited to solve optimization problems, according to the University of Massachusetts, Lowell Center for Atmospheric Research. Feedback ANNs are used by the Internal system error corrections.
- **Feed Forward ANN –** A feed-forward network is a simple neural network consisting of an input layer, an output layer and one or more layers of neurons.Through evaluation of its output by reviewing its input, the power of the network can be noticed base on group behavior of the connected neurons and the output is decided. The main advantage of this network is that it learns to evaluate and recognize input patterns.
- **Classification-Prediction ANN** –It is the subset of feed-forward ANN and the classification-prediction ANN is applied to data-mining scenarios. The network is trained to identify particular patterns and classify them into specific groups and then further classify them into "novel patterns" which are new to the network

7. **Distinguish between Learning and Training.**

The training function is the overall algorithm that is used to train the neural network to recognize a certain input and map it to an output. A common example is backpropagation and its many variations and weight/bias training.

A learning function deals with individual weights and thresholds and decides how those would be manipulated. These usually (but not always) employ some form of gradient descent. Examples include simulated annealing, Silva and Almeida's algorithm, using momentum and adaptive learning-rates, and weight-learning (examples include Hebb, Kohonen, etc.) algorithms.

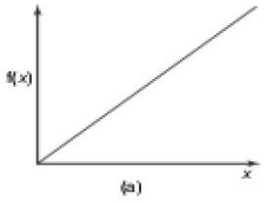
8. **How can you measure the similarity of two patterns in the input space**?

9. Mention the linear and nonlinear activation functions used in Artificial neural networks.

1.*Linear Activation Function/Identity function*

1. *Identity function*: It is a linear function and can be defined as

$$f(x) = x \quad \text{for all } x$$



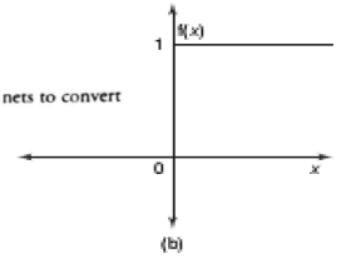
2. Non-linear Activation function

I. Binary Step

2. *Binary step function*: This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$$

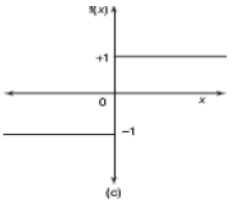
where θ represents the threshold value. This function is most widely used in single-layer nets to convert the net input to an output that is a binary (1 or 0).



3. *Bipolar step function*: This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$

where θ represents the threshold value. This function is also used in single-layer nets to convert the net input to an output that is bipolar (+1 or −1).



4. *Sigmoidal functions*: The sigmoidal functions are widely used in back-propagation nets because of the relationship between the value of the functions at a point and the value of the derivative at that point which reduces the computational burden during training.
Sigmoidal functions are of two types:

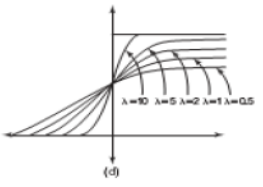
- *Binary sigmoid function*: It is also termed as logistic sigmoid function or unipolar sigmoid functic It can be defined as

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

where λ is the steepness parameter. The derivative of this function is

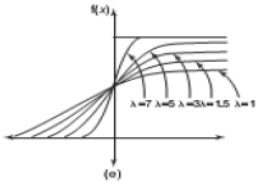
$$f'(x) = \lambda f(x)[1 - f(x)]$$

Here the range of the sigmoid function is from 0 to 1.



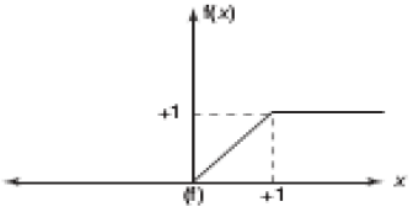
- *Bipolar sigmoid function*: This function is defined as

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



(F) Ramp

$$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } 0 \le x \le 1 \\ 0 & \text{if } x < 0 \end{cases}$$



10. Write the differences between conventional computers and ANN.

Parallel processing

One of the major advantages of the neural network is its ability to do many things at once. With traditional computers, processing is sequential--one task, then the next, then the next, and so on. The idea of threading makes it appear to the human user that many things are happening at one time. For instance, the Netscape throbber is shooting meteors at the same time that the page is loading. However, this is only an appearance; processes are not actually happening simultaneously.

The artificial neural network is an inherently multiprocessor-friendly architecture. Without much modification, it goes beyond one or even two processors of the von Neumann architecture. The artificial neural network is designed from the onset to be parallel. Humans can listen to music at the same time they do their homework--at least, that's what we try to convince our parents in high school. With a massively parallel architecture, the neural network can accomplish a lot in less time. The tradeoff is that processors have to be specifically designed for the neural network.

The ways in which they function

Another fundamental difference between traditional computers and artificial neural networks is the way in which they function. While computers function logically with a set of rules and calculations, artificial neural networks can function via images, pictures, and concepts.

Based upon the way they function, traditional computers have to learn by rules, while artificial neural networks learn by example, by doing something and then learning from it. Because of these fundamental differences, the applications to which we can tailor them are extremely different. We will explore some of the applications later in the presentation.

Self-programming

The "connections" or concepts learned by each type of architecture is different as well. The von Neumann computers are programmable by higher level languages like C or Java and then translating that down to the machine's assembly language. Because of their style of learning, artificial neural networks can, in essence, "program themselves." While the conventional computers must learn only by doing different sequences or steps in an algorithm, neural networks are continuously adaptable by truly altering their own programming. It could be said that conventional computers are limited by their parts, while neural networks can work to become more than the sum of their parts.

Speed

The speed of each computer is dependant upon different aspects of the processor. Von Neumann machines requires either big processors or the tedious, error-prone idea of parallel processors, while neural networks requires the use of multiple chips customly built for the application.

11. Explain in Detail how weights are adjusted in the different types of Learning Law. (Both supervised and Unsupervised)

Supervised learning is based on weight adjustment **based on deviation of desired output from actual output**.

12. **Write short notes on the following.**

A. ***Learning Rate Parameter***

Learning rate is **a hyper-parameter that controls the weights of our neural network with respect to the loss gradient**. It defines how quickly the neural network updates the concepts it has learned.

A desirable learning rate is low enough that the network converges to something useful, but high enough that it can be trained in a reasonable amount of time.

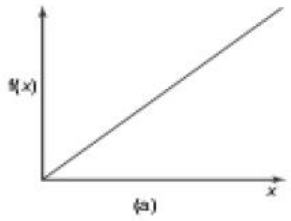
Smaller learning rates require more training epochs (requires more time to train) due to the smaller changes made to the weights in each update, whereas larger learning rates result in rapid changes and require fewer training epochs. However, larger learning rates often result in a sub-optimal final set of weights.

B. **Activation Function**.

An Activation Function **decides whether a neuron should be activated or not**. This means that it will decide whether the neuron's input to the network is important or not in the process of prediction using simpler mathematical operations.

1. ***Identity function:*** **It is a linear function and can be defined as**

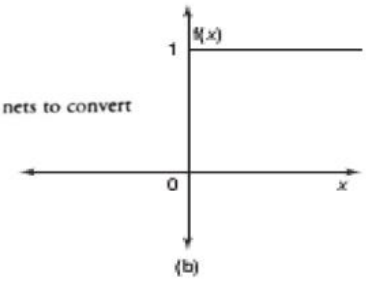
$$f(x) = x \quad \text{for all } x$$



2. *Binary step function*: This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$$

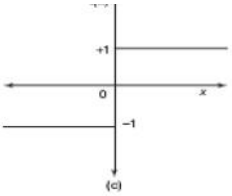
where θ represents the threshold value. This function is most widely used in single-layer nets to convert the net input to an output that is a binary (1 or 0).



3. *Bipolar step function*: This function can be defined as

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ -1 & \text{if } x < \theta \end{cases}$$

where θ represents the threshold value. This function is also used in single-layer nets to convert the net input to an output that is bipolar (+1 or −1).



(c)

4. *Sigmoidal functions*: The sigmoidal functions are widely used in back-propagation nets because of the relationship between the value of the functions at a point and the value of the derivative at that point which reduces the computational burden during training.
Sigmoidal functions are of two types:

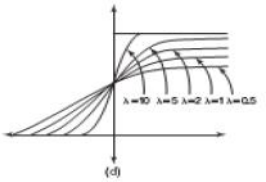
- *Binary sigmoid function*: It is also termed as logistic sigmoid function or unipolar sigmoid functic It can be defined as

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

where λ is the steepness parameter. The derivative of this function is

$$f'(x) = \lambda f(x)[1 - f(x)]$$

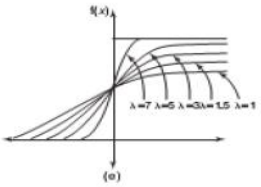
Here the range of the sigmoid function is from 0 to 1.



(d)

- *Bipolar sigmoid function*: This function is defined as

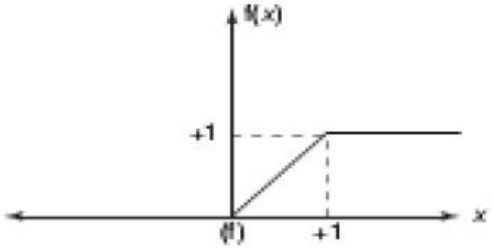
$$f(x) = \frac{2}{1 + e^{-\lambda x}} - 1 = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}}$$



(e)

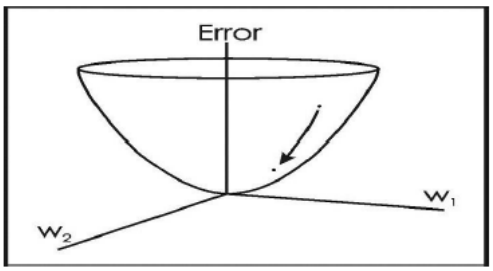
(F) Ramp

$$f(x) = \begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x < 0 \end{cases}$$



12.Explain Delta Learning Rule.

DELTA LEARNING RULE



For any given set of input data and weights, there will be an associated magnitude of error, which is measured by an error function (also known as a cost function). The Delta Rule employs the error function for what is known as Gradient Descent learning, which involves the *'modification of weights along the most direct path in weight-space to minimize error', so* change applied to a given weight is proportional to the negative of the derivative of the error with respect to that weight

$$E_p = \frac{1}{2}\sum_n (t_{j_n} - a_{j_n})^2$$

Delta Learning Rule

The delta learning rule is only valid for continuous activation functions.

$$f(net) = \frac{2}{1+\exp^{-\lambda net}} - 1 \quad \text{— Bipolar}$$

or

$$= \frac{1}{1+\exp^{-\lambda net}} \quad \text{— Unipolar}$$

The learning signal for this rule is called delta and is defined as follows:-

$$r = [d_i - \underbrace{f(w_i^t x)}_{o_i}]\, f'(w_i^t x) \quad \text{— (1)}$$

The term $f'(w_i^t x)$ is the derivative of the activation function $f(net)$ computed for $net = w_i^t x$.

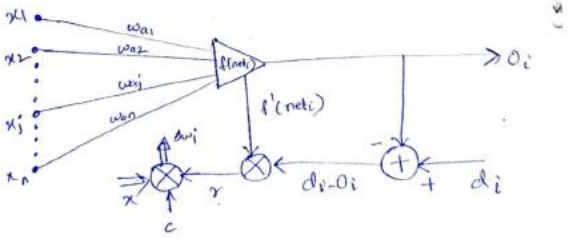


Fig: Delta Learning Rule

This learning rule can be readily derived from the condition of least squared error between o_i and d_i. Calculating the gradient vector with respect to w_i of the squared error defined as,

$$E = \frac{1}{2}(d_i - o_i)^2 \quad \text{---(2)}$$

or

$$E = \frac{1}{2}[d_i - f(w_i^t x)]^2 \quad \text{---(3)}$$

we obtain the error gradient ~~vector~~, vector value,

$$\nabla E = -(d_i - o_i) f'(w_i^t x) \cdot x \quad \text{---(4)}$$

The components of the gradient vector are,

$$\frac{\partial E}{\partial w_{ij}} = -(d_i - o_i) f'(w_i^t x) x_j \quad \text{for } j = 1, 2, \ldots n \quad \text{---(5)}$$

Since, minimization of the error requires the weight changes to be in the negative gradient direction we take

$$\Delta w_i = -\eta \nabla E \quad \text{---(6)}$$

where η is a positive constant

From Eq. (4) & (6) we obtain,

$$\Delta w_i = -\eta (d_i - o_i) f'(net_i) \cdot x \quad \text{---(7)}$$

or, for the single weight the adjustment becomes

$$\Delta w_{ij} = \eta (d_i - o_i) f'(net_i) x_j \quad \text{---(8)}$$

for $j = 1, 2, \ldots n$

Considering the use of general learning rule and plugging in the learning signal, the weight adjustment becomes,

$$\Delta w_i = c (d_i - o_i) f'(net_i) \cdot x \quad \text{---(9)}$$

From equations (4), (8) & (9) we can conclude that both are identical as, c of otherhave been assumed to be arbitrary constants.

⇒ The weights are initialized at any values for this method of training.

⇒ This rule parallels the discrete perception training rule. It can also be called as continuous perception training rule. The Delta learning rule can be generalized for multilayer networks.

$f'(net) = \frac{1}{2}(d_i^2 - o_i^2)$

13. Briefly describe Perceptron Learning Rule.

Perceptron Learning Rule

For the perceptron learning rule signal is the difference between the desired and actual neuron's response. Thus, learning is supervised and the learning signal is equal to,

$$r = d_i - O_i$$

where $O_i = \text{sgn}(w_i^t x)$ and d_i is the desired response.

Weight adjustments in this method, Δw_i and Δw_{ij} are obtained as follows:-

$$\Delta w_i = c[d_i - \underbrace{\text{sgn}(w_i^t x)}_{O_i}]x$$

$$\Delta w_{ij} = c[d_i - \text{sgn}(w_i^t x)]x_j \quad \text{for } j = 1, 2, \ldots$$

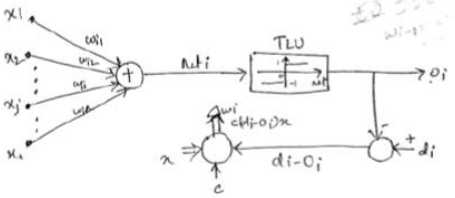
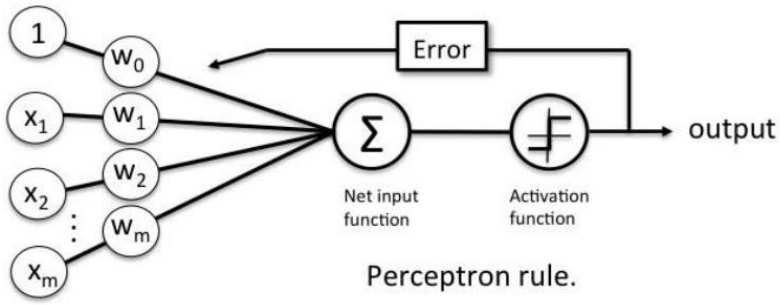


Fig: Perceptron Learning Rule



Perceptron rule.

⇒ This rule is applicable for binary neuron response. i.e Rule for the binary bipolar

⇒ Under this rule weights are adjusted if and only if O_i is incorrect.

⇒ As the desired response is +1 or −1 the weight adjustment (2) reduces to,

$$\Delta w_i = \pm 2cx \quad \text{---} \quad (4)$$

⇒ where a + is applicable when $d_i = 1$ and $\text{Sgn}(w^t x) = -1$ and a minus sign is applicable when $d_i = -1$ and $\text{sgn}(w^t x) = 1$.

⇒ The wt. adjustment formula will not be used when $d_i = \text{sgn}(w_i^t x)$

⇒ This is a very important rule for supervised Learning Rule.

⇒ The weight are initialized at any value in this method.

Algorithm: Perceptron Learning Algorithm

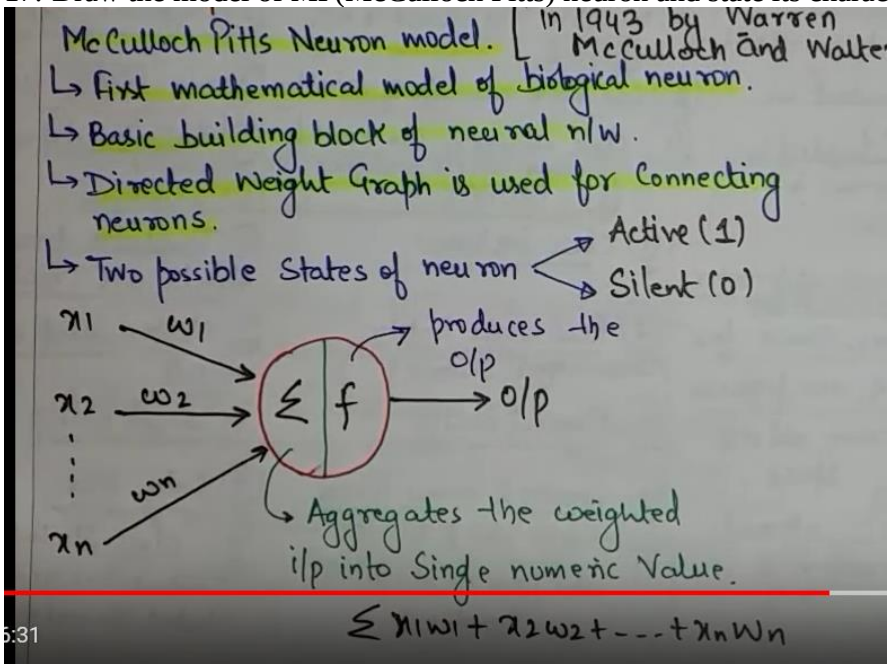
$P \leftarrow$ *inputs with label* 1;
$N \leftarrow$ *inputs with label* 0;
Initialize **w** randomly;
while !*convergence* **do**
  Pick random $\mathbf{x} \in P \cup N$;
  if $\mathbf{x} \in P$ *and* $\sum_{i=0}^{n} w_i * x_i < 0$ **then**
    $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
  end
  if $\mathbf{x} \in N$ *and* $\sum_{i=0}^{n} w_i * x_i \geq 0$ **then**
    $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
  end
end
//the algorithm converges when all the inputs are classified correctly

14. What are the relevant computational properties of the Human Brain.

15. Distinguish between linearly separable and nonlinearly separable problems. Give examples.

16. Compare physical neuron and artificial neuron.

17. Draw the model of MP(McCulloch Pitts) neuron and state its characteristics.



18. Draw the structure of a biological Neuron and explain in detail

.

19. Draw the architecture of a single layer perceptron (SLP) and explain its operation. Mention its advantages and disadvantages.

20. Develop simple ANNs to implement the three input AND, OR and XOR functions using MP neurons.

22. Explain different architectures of Neural Network.