1. What is the function of a summation junction of a neuron? What is threshold activation function?
2. ) A neuron's summation junction is in responsible for merging any weighted inputs it receives. subsequently computing the inputs' weighted sum, the activation function is applied. On the other hand, based upon whether the derived sum surpasses an initial threshold, the threshold activation function chooses whether the neuron will fire.
3. What is a step function? What is the difference of step function with threshold function?
4. ) A mathematical function called a step function, sometimes referred to as a Heaviside step function, returns 0 or 1 depending on whether the input is less than or equal to zero. A threshold function compares an input value with a threshold before determining the output, whereas a step function outputs 0 or 1. This is the primary distinction between a threshold function and a step function.
5. Explain the McCulloch–Pitts model of neuron.
6. ) A simplified model of a neuron that depicts a biological neuron is called the McCulloch-Pitts model. It takes in binary inputs, weights them, adds them up, and then outputs binary (0 or 1) based on passing the result through a threshold function.
7. Explain the ADALINE network model.
8. ) A single-layer artificial neural network called ADALINE (Adaptive Linear Neuron) is used to carry out linear regression. By utilizing the Widrow-Hoff learning rule, it modifies its weights according to the discrepancy between the output and the intended output.

1. What is the constraint of a simple perceptron? Why it may fail with a real-world data set?

A.) A simple perceptron's learning ability is limited to functions that are linearly separable. If real-world data sets are not linearly separable—that is, if there isn't a single straight line that can divide the classes—it might not work.

1. What is linearly inseparable problem? What is the role of the hidden layer?
2. ) Problems that are not separable by a single line are known as linearly inseparable problems. In order for a neural network to learn non-linear relationships and patterns and solve more complicated problems, it needs a hidden layer.
3. Explain XOR problem in case of a simple perceptron.
4. ) The XOR problem occurs when one tries to learn the XOR function—which is not linearly separable—using a basic perceptron. Because XOR is linear, a single-layer perceptron is unable to learn it.
5. Design a multi-layer perceptron to implement A XOR B.
6. ) A multi-layer perceptron would require at least one hidden layer in order to implement A XOR B. The network is able to grasp the XOR function's non-linearity thanks to the hidden layer.
7. Explain the single-layer feed forward architecture of ANN.
8. ) Information moves from input to output in a single direction, without cycles, in an Artificial Neural Network (ANN) with a single-layer feedforward architecture. There are no feedback connections between any two neurons in the layer and every other neuron in the layer below it.
9. Explain the competitive network architecture of ANN.
10. ) Neurons in an ANN compete with one another to become active in its competitive network architecture. The competition winner is represented by the only neuron that fires at a time. Clustering is one common use for this kind of network.
11. Consider a multi-layer feed forward neural network. Enumerate and explain steps in the backpropagation algorithm used to train the network.
12. ) The forward pass, error calculation, backward pass, gradient descent algorithm weight updates, and iteration over several epochs to refine the weights are all steps in the backpropagation algorithm used to train a multi-layer feedforward neural network.
13. What are the advantages and disadvantages of neural networks?
14. ) Neural networks have the capacity to learn intricate patterns, to adapt to different tasks, and to process information in parallel. Large dataset requirements, high processing demands, and the risk of overfitting are drawbacks.
15. Write short notes on any two of the following:
    * 1. Biological neuron
      2. **ReLU function**

**A.)** Artificial neural networks frequently use the activation function ReLU. By directly outputting the input if it is positive and zero otherwise, it adds non-linearity to the network. ReLU can be defined mathematically as f(x) = max(0, x). ReLU's simplicity helps it overcome the vanishing gradient issue that other activation functions have, as well as making it computationally efficient. Nevertheless, ReLU might have the "dying ReLU" issue, in which neurons might stop learning and become dormant during training. This problem is addressed by variants such as Leaky ReLU, which permit a small, non-zero gradient for negative inputs.

* + 1. **Single-layer feed forward ANN**

1. **)** The most basic type of neural network architecture is a single-layer feedforward artificial neural network (ANN). It is made up of an input layer, an output layer, and a single layer of neurons that serve as processing units. Without any feedback loops or hidden layers, information moves directly from the input layer to the output layer. Each neuron's output is the result of applying an activation function after calculating the weighted sum of its inputs. Because single-layer feedforward networks don't have any hidden layers, they can only solve linearly separable problems. Although they served as a foundation for the development of neural networks, multi-layer networks are better able to handle complex, non-linear relationships.
   * 1. Gradient descent
     2. Recurrent networks