**1. Describe the structure of an artificial neuron. How is it similar to a biological neuron? What are its main components?**

* **Structure of an artificial neuron**:
  + **Inputs**: Receive input signals (e.g., x1,x2,…,xnx1​,x2​,…,xn​).
  + **Weights**: Each input is multiplied by a weight (w1,w2,…,wnw1​,w2​,…,wn​).
  + **Summation junction**: Computes the weighted sum of inputs (z=∑i=1nwixi+bz=∑i=1n​wi​xi​+b), where bb is the bias.
  + **Activation function**: Applies a non-linear function (e.g., sigmoid, ReLU) to the weighted sum to produce the output.
* **Similarity to a biological neuron**:
  + Both receive inputs (dendrites in biological neurons), process them (cell body), and produce an output (axon).
  + The weights in an artificial neuron are analogous to synaptic strengths in biological neurons.
* **Main components**:
  + Inputs, weights, summation junction, bias, and activation function.

**2. What are the different types of activation functions popularly used? Explain each of them.**

* **Step function**: Outputs 1 if the input is above a threshold, otherwise 0. Used in binary classification.
* **Sigmoid function**: f(x)=11+e−xf(x)=1+e−x1​. Outputs values between 0 and 1, useful for probability estimation.
* **ReLU (Rectified Linear Unit)**: f(x)=max⁡(0,x)f(x)=max(0,x). Widely used in deep learning due to its simplicity and efficiency.
* **Tanh (Hyperbolic Tangent)**: f(x)=ex−e−xex+e−xf(x)=ex+e−xex−e−x​. Outputs values between -1 and 1, often used in hidden layers.
* **Softmax**: Used in the output layer for multi-class classification. Converts raw scores into probabilities.

**3. Explain Rosenblatt's perceptron model. How can a set of data be classified using a simple perceptron?**

* **Rosenblatt's perceptron**:
  + A single-layer neural network with binary inputs and outputs.
  + Uses a step function as the activation function.
  + Learns weights using the perceptron learning rule: weights are updated based on the error between the predicted and actual output.
* **Classifying data**:
  + The perceptron computes the weighted sum of inputs and applies a threshold to classify the data.
  + For example, with weights w0=−1w0​=−1, w1=2w1​=2, and w2=1w2​=1, the perceptron classifies data points by computing z=w0+w1x1+w2x2z=w0​+w1​x1​+w2​x2​ and applying a step function.

**4. Explain the basic structure of a multi-layer perceptron. How can it solve the XOR problem?**

* **Structure**:
  + Consists of an input layer, one or more hidden layers, and an output layer.
  + Each layer contains multiple neurons with non-linear activation functions.
* **Solving XOR**:
  + A single-layer perceptron cannot solve XOR because it is linearly inseparable.
  + A multi-layer perceptron with at least one hidden layer can learn non-linear decision boundaries, enabling it to solve XOR.

**5. What is an artificial neural network (ANN)? Explain some of the salient highlights in the different architectural options for ANN.**

* **ANN**:
  + A computational model inspired by biological neural networks, consisting of interconnected neurons.
* **Architectural options**:
  + **Feedforward networks**: Information flows in one direction (input to output).
  + **Recurrent networks**: Allow feedback loops, useful for sequential data.
  + **Convolutional networks**: Specialized for image processing, using convolutional layers.
  + **Autoencoders**: Used for unsupervised learning and dimensionality reduction.

**6. Explain the learning process of an ANN. Explain, with example, the challenge in assigning synaptic weights for the interconnection between neurons. How can this challenge be addressed?**

* **Learning process**:
  + Involves adjusting weights to minimize the error between predicted and actual outputs.
  + Uses optimization algorithms like gradient descent.
* **Challenge**:
  + Assigning weights manually is impractical for large networks.
* **Solution**:
  + Use backpropagation to automatically adjust weights based on the error gradient.

**7. Explain, in details, the backpropagation algorithm. What are the limitations of this algorithm?**

* **Backpropagation**:
  + A method for training ANNs by propagating errors backward through the network.
  + Steps:
    1. Forward pass: Compute the output.
    2. Calculate error: Compare output with target.
    3. Backward pass: Compute gradients for each weight.
    4. Update weights: Adjust weights using gradient descent.
* **Limitations**:
  + Prone to getting stuck in local minima.
  + Computationally expensive for large networks.

**8. Describe, in details, the process of adjusting the interconnection weights in a multi-layer neural network.**

* **Process**:
  + Initialize weights randomly.
  + Compute the output using forward propagation.
  + Calculate the error between predicted and actual output.
  + Use backpropagation to compute gradients for each weight.
  + Update weights using gradient descent: wnew=wold−η∂E∂wwnew​=wold​−η∂w∂E​, where ηη is the learning rate.

**9. What are the steps in the backpropagation algorithm? Why is a multi-layer neural network required?**

* **Steps**:
  1. Forward pass: Compute the output.
  2. Calculate error: Compare output with target.
  3. Backward pass: Compute gradients for each weight.
  4. Update weights: Adjust weights using gradient descent.
* **Why multi-layer networks are required**:
  1. Single-layer networks can only solve linearly separable problems. Multi-layer networks can model complex, non-linear relationships.

**10. Write short notes on:**

1. Artificial neuron:

A computational unit that mimics a biological neuron, consisting of inputs, weights, a summation junction, and an activation function.

2. Multi-layer perceptron:

A neural network with one or more hidden layers, capable of solving non-linear problems.

3. Deep learning:

A subfield of machine learning that uses deep neural networks with many layers to model complex patterns in data.

4. Learning rate:

A hyperparameter that controls the step size during weight updates in gradient descent. Too high can cause instability; too low can slow convergence.