Q1)**Architecture of Spark**:

Apache Spark follows a master-slave architecture, consisting of the following components:
Driver: The driver is the central component that manages the execution of the Spark application. It converts user code into tasks and coordinates their execution on the cluster.
Cluster Manager: Spark can run on various cluster managers like YARN, Mesos, or its standalone cluster manager. The cluster manager allocates resources and manages the execution of tasks across the cluster.
Worker Nodes: Worker nodes are the machines in the cluster that execute the tasks. They receive instructions from the driver and report back their status.
Executor: Executors are processes launched on worker nodes to execute tasks. Each application has its set of executors, which stay alive for the duration of the Spark application.
Task: Tasks are units of work that perform computations and are executed by the executors. They are spawned for each partition of the RDD (Resilient Distributed Dataset).
RDD (Resilient Distributed Dataset): RDD is the fundamental data structure in Spark, representing a collection of immutable, partitioned records that can be operated on in parallel.

Q2)**Activation Function**:

In neural networks, an activation function is a mathematical function that determines the output of a neuron. It introduces non-linearity into the network, allowing it to learn complex patterns in the data. The activation function operates on the weighted sum of inputs to the neuron (plus a bias term) and produces the output of the neuron.

Q3) some common activation functions used in neural networks along with their formulas:
1. Sigmoid Activation Function:

Formula

$\frac{1}{1+e^{-x}}$

Range: (0, 1)

$$\sigma'(x) = \sigma(x) \cdot (1 - \sigma(x))$$

2. **Tanh** (Hyperbolic Tangent) Activation Function:

Formula: $: \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

Range: (-1, 1)

Derivative:

$\text{sech}^2(x)$ (where $\text{sech}(x) = \frac{1}{\cosh(x)}$)

3. **ReLU** (Rectified Linear Unit) Activation Function:

Formula:

$$f(x) = \max(0, x)$$

Range: [0, ∞)

Derivative:

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

4. **Leaky ReLU Activation Function**:

Formula

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha \cdot x, & \text{otherwise} \end{cases}$$

Range: (-∞, ∞)

Derivative:

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ \alpha, & \text{otherwise} \end{cases}$$

5. **Softmax Activation Function**:

Formula:

$\mathrm{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j-1}^{N} e^{x_j}}$ for i = 1, ..., N where N is the number of classes

Range: (0, 1) for each element, with the sum of all elements being 1

6. **Linear Activation Function**:

Formula:

$$f(x) = x$$

Range: (-∞, ∞)

Derivative:

$$f'(x) = 1$$

Q5) Neural networks are a class of machine learning models inspired by the structure and function of the human brain. They are composed of interconnected nodes, called neurons, which process and transmit information. Neural networks are used for various tasks, including classification, regression, and clustering

Components of a Neural Network:

1. Input Layer:The input layer receives the initial data or features to be processed by the neural network.

2. Hidden Layers: Hidden layers are layers between the input and output layers where the computation and transformation of data occur. Each neuron in a hidden layer processes information from the previous layer and passes it on to the next layer.

3.Output Layer: The output layer produces the final output of the neural network based on the computations performed in the hidden layers. The number of neurons in the output layer depends on the type of problem being solved (e.g., binary classification, multi-class classification, regression).

4. Weights and Biases: Each connection between neurons in adjacent layers has an associated weight, which determines the strength of the connection. Biases are added to the weighted sum of inputs to introduce flexibility in the model.

5 Activation Function Activation functions introduce non-linearity into the neural network, allowing it to learn complex patterns in the data. Common activation functions include sigmoid, tanh, ReLU, and softmax.

6. Loss Function: The loss function measures how well the neural network's predictions match the actual target values. It is used to adjust the weights and biases during the training process to minimize the error.

7. Optimizer: The optimizer is used to update the weights and biases of the neural network based on the gradients of the loss function. Popular optimizers include Stochastic Gradient Descent (SGD), Adam, and RMSprop.

Training a Neural Network:

Training a neural network involves feeding it with input data and adjusting the weights and biases through a process called backpropagation. During training, the network learns to minimize the loss function by updating its parameters based on the error between the predicted output and the actual output.

Applications of Neural Networks:

Neural networks are used in a wide range of applications, including:

- Image and speech recognition
- Natural language processing
- Handwriting recognition
- Autonomous vehicles
- Financial forecasting
- Healthcare diagnostics

Neural networks have demonstrated high accuracy and effectiveness in handling complex, large-scale datasets, making them a powerful tool in modern machine learning and artificial intelligence.