1. Architecture of Spark:

Apache Spark is an open-source, distributed computing system that provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Its architecture consists of the following key components:

- Driver: The central coordinator that schedules tasks and manages resources across the Spark application.

- Cluster Manager: Manages resources across the Spark cluster (e.g., YARN, Mesos, Kubernetes).

- Worker Nodes: Executors that run tasks and store data for the Spark application.

- Executor: Processes that execute tasks, responsible for in-memory computing and storage.

- Task: Units of work that perform computations and return results to the driver.

-RDD (Resilient Distributed Dataset): Immutable distributed collections of objects partitioned across nodes in the cluster. RDDs are the fundamental data structure in Spark.

- DAG (Directed Acyclic Graph): Represents the sequence of computations as a graph, where vertices are RDDs and edges are operations.

2.Activation Function:

An activation function in neural networks defines the output of a node given an input or set of inputs. It introduces non-linearity to the model, enabling it to learn complex patterns in the data. Common activation functions include:

Sigmoid: Maps input values to a range between 0 and 1, suitable for binary classification tasks.

-ReLU (Rectified Linear Unit): Sets negative values in the input to zero and leaves positive values unchanged, helping with the vanishing gradient problem.

Tanh: Similar to the sigmoid function but maps input values to a range between -1 and 1, useful for outputs that can be negative.

Activation functions play a crucial role in determining the output of a neural network and are essential for learning complex patterns in data.

### 3. Different Types of Activation Functions with Their Formulas:

1. Sigmoid Function:

- Formula: \( \sigma(x) = \frac{1}{1 + e^{-x}} \)

- Range: (0, 1)

- Use: Mostly used in the output layer of binary classification problems.

2. Tanh (Hyperbolic Tangent) Function:

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

- Range: (-1, 1)

- Use: Commonly used in hidden layers of neural networks.

3. ReLU (Rectified Linear Activation) Function:

- Formula: \( f(x) = \max(0, x) \)

- Range: [0, ∞)

- Use: Very popular in hidden layers due to its simplicity and effectiveness in training deep neural networks.

4. Leaky ReLU Function:

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

- Range: (-∞, ∞)

- Use: Addresses the "dying ReLU" problem by allowing a small, non-zero gradient when x < 0.

5. Softmax Function:

- Formula: \( \text{softmax}(x\_{i}) = \frac{e^{x\_{i}}}{\sum\_{j=1}^{N} e^{x\_{j}}} \) for i = 1, ..., N

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

- Use: Typically used in the output layer of a neural network for multi-class classification, as it converts the raw scores (logits) into probabilities.

Q5. Explanation of Neural Networks:

Neural networks are computational models inspired by the structure and function of biological neural networks in the human brain. They consist of interconnected nodes organized into layers, each performing specific computations. Here's a breakdown of key components:

- Neuron (Node): The basic building block of a neural network. Each neuron receives input signals, applies a transformation to the input using weights and biases, and produces an output signal.

- Layers: Neurons are organized into layers. The input layer receives the initial data, the output layer produces the final output, and the intermediate layers are called hidden layers, where most of the computation occurs.

- Connections: Neurons in adjacent layers are connected by weighted connections. Each connection has an associated weight that determines its strength. During training, these weights are adjusted to minimize the error in the network's predictions.

- Activation Function: Each neuron applies an activation function to the weighted sum of its inputs to introduce non-linearity into the network. This allows neural networks to learn complex patterns in the data.

- Feedforward Propagation: The process of passing the input data through the network to produce predictions or outputs. It involves computing the output of each neuron layer by layer, from the input layer to the output layer.

- Backpropagation: The process of training the neural network by adjusting the weights and biases based on the error between the predicted output and the actual output. It involves computing the gradients of the loss function with respect to the network parameters and updating the parameters using gradient descent or its variants.

Neural networks have shown remarkable success in various tasks such as image recognition, speech recognition, natural language processing, and many more. They are powerful tools for learning from data and making predictions in complex, real-world problems.