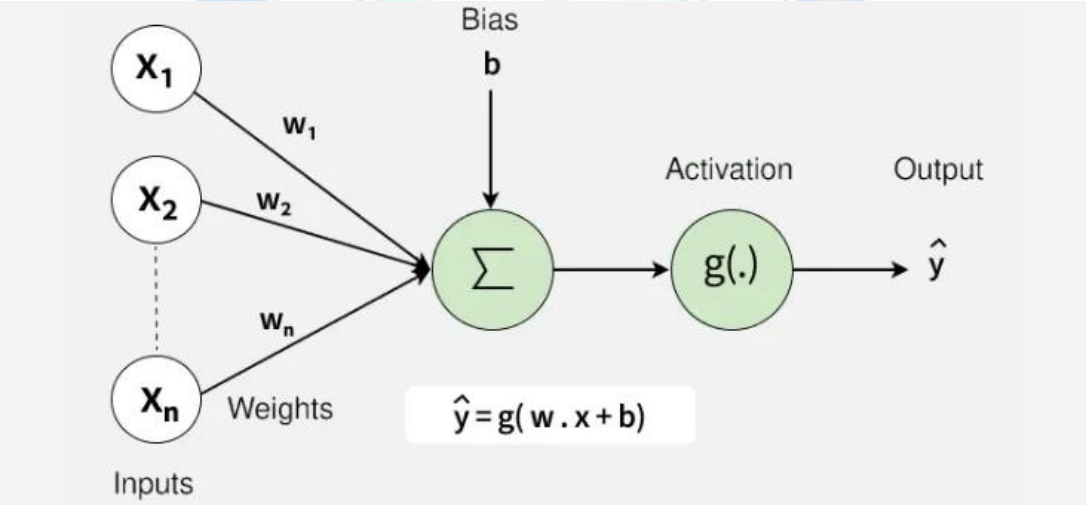
NEURAL NETWORKS

Neural Networks are machine learning models inspired by the way the human brain processes information. They consist of interconnected layers of neurons that learn patterns from data and make predictions.

- Learn complex patterns from large datasets.
- Consist of input, hidden, and output layers.
- Used in image recognition, text processing, speech recognition, and prediction tasks.



BASIC COMPONENTS OF A NEURAL NETWORK

A neural network consists of the following important components:

1. Neuron

A **neuron** is the basic processing unit of a neural network. It receives input values, performs a mathematical computation, and produces an output.

2. Connections

Connections link one neuron to another and allow information to pass between neurons.

3. Weights

A **weight** represents the importance or strength of an input connection.

A larger weight means that the corresponding input has a greater influence on the neuron output.

4. Bias

A **bias** is an additional parameter added to the weighted sum. It helps the neuron adjust its output independently of the input values.

5. Activation Function

The activation function determines whether and how strongly a neuron should produce an output. It also introduces **non-linearity**, allowing the network to learn complex patterns.

Common activation functions:

- ReLU
- Sigmoid
- Tanh

6. Learning Rule

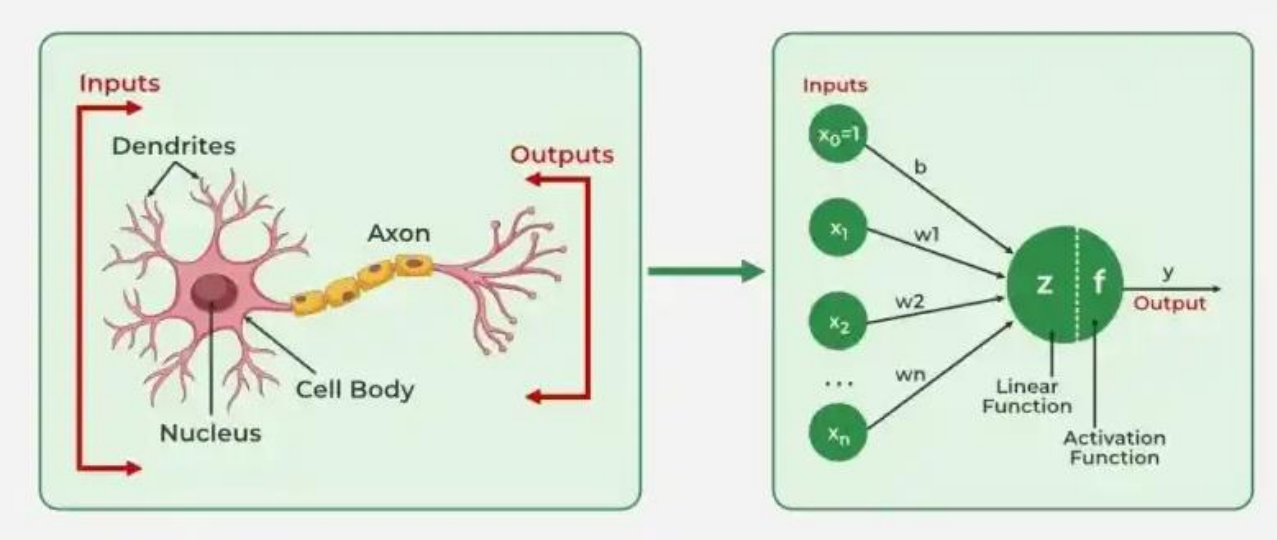
The learning rule determines how the weights and biases are modified during training to reduce prediction errors.

Learning in neural networks follows a structured, three-stage process:

1. **Input Computation**: Data is fed into the network.
2. **Output Generation**: Based on the current parameters, the network generates an output.
3. **Iterative Refinement**: The network refines its output by adjusting weights and biases, gradually improving its performance on diverse tasks.

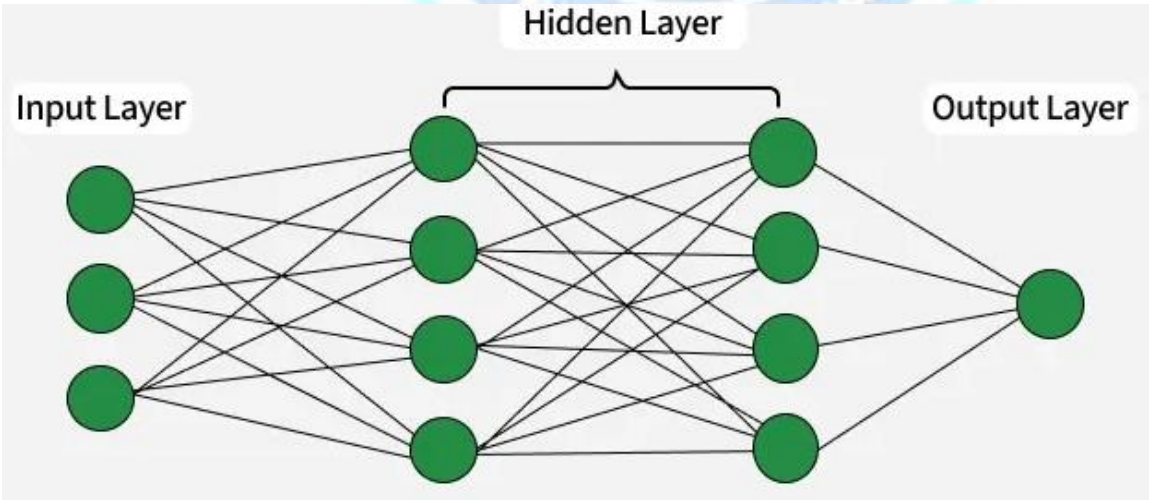
Neural Network Training Process

Input → Forward Propagation → Output → Loss Calculation → Backpropagation → Weight Update → Repeat



The image illustrates the analogy between a biological neuron and an artificial neuron, showing how inputs are received and processed to produce outputs in both systems.

Layers in Neural Network Architecture



Layers in Neural Network Architecture

1. **Input Layer:** This is where the network receives its input data. Each input neuron in the layer corresponds to a feature in the input data.

2. **Hidden Layers:** These layers perform most of the computational heavy lifting. A neural network can have one or multiple hidden layers. Each layer consists of units (neurons) that transform the inputs into something that the output layer can use.
3. **Output Layer:** The final layer produces the output of the model. The format of these outputs varies depending on the specific task like classification, regression.

WORKING OF NEURAL NETWORKS

1. Forward Propagation

When data is input into the network, it passes through the network in the forward direction, from the input layer through the hidden layers to the output layer. This process is known as forward propagation. Here's what happens during this phase:

1. Linear Transformation: Each neuron in a layer receives inputs which are multiplied by the weights associated with the connections. These products are summed together and a bias is added to the sum. This can be represented mathematically as:

z=w1x1+w2x2+…+wnxn+bz=*w*1*x*1+*w*2*x*2+…+*wnxn*+*b*

where

- w*w* represents the weights
- x*x* represents the inputs
- b*b* is the bias

2. Activation: The result of the linear transformation (denoted as z*z*) is then passed through an activation function. The activation function is crucial because it introduces non-linearity into the system, enabling the network to learn more complex patterns. Popular activation functions include ReLU, sigmoid and tanh.

2. Backpropagation

After forward propagation, the network evaluates its performance using a loss function which measures the difference between the actual output and the predicted output. The goal of training is to minimize this loss. This is where backpropagation comes into play:

- **Loss Calculation:** The network calculates the loss which provides a measure of error in the predictions. The loss function could vary; common choices are mean squared error for regression tasks or cross-entropy loss for classification.
- **Gradient Calculation:** The network computes the gradients of the loss function with respect to each weight and bias in the network. This involves applying the

chain rule of calculus to find out how much each part of the output error can be attributed to each weight and bias.

- **Weight Update:** Once the gradients are calculated, the weights and biases are updated using an optimization algorithm like stochastic gradient descent (SGD). The weights are adjusted in the opposite direction of the gradient to minimize the loss. The size of the step taken in each update is determined by the learning rate.

3. Iteration

This process of forward propagation, loss calculation, backpropagation and weight update is repeated for many iterations over the dataset. Over time, this iterative process reduces the loss and the network's predictions become more accurate.

Through these steps, neural networks can adapt their parameters to better approximate the relationships in the data, thereby improving their performance on tasks such as classification, regression or any other predictive modeling.

Learning of a Neural Network

1. Learning with Supervised Learning

In supervised learning, a neural network learns from labeled input-output pairs provided by a teacher. The network generates outputs based on inputs and by comparing these outputs to the known desired outputs, an error signal is created. The network iteratively adjusts its parameters to minimize errors until it reaches an acceptable performance level.

2. Learning with Unsupervised Learning

Unsupervised learning involves data without labeled output variables. The primary goal is to understand the underlying structure of the input data (X). Unlike supervised learning, there is no instructor to guide the process. Instead, the focus is on modeling data patterns and relationships, with techniques like clustering and association commonly used.

3. Learning with Reinforcement Learning

Reinforcement learning enables a neural network to learn through interaction with its environment. The network receives feedback in the form of rewards or penalties, guiding it to find an optimal policy or strategy that maximizes cumulative rewards over time. This approach is widely used in applications like gaming and decision-making.

Neural networks are available in different types, each designed for specific tasks and data types. For more details, refer to Types of Neural Networks.

Applications

1. **Image and Video Recognition**: CNNs are extensively used in applications such as facial recognition, autonomous driving and medical image analysis.
2. **Natural Language Processing (NLP)**: RNNs and transformers power language translation, chatbots and sentiment analysis.
3. **Healthcare**: Neural networks assist in diagnosing diseases, analyzing medical