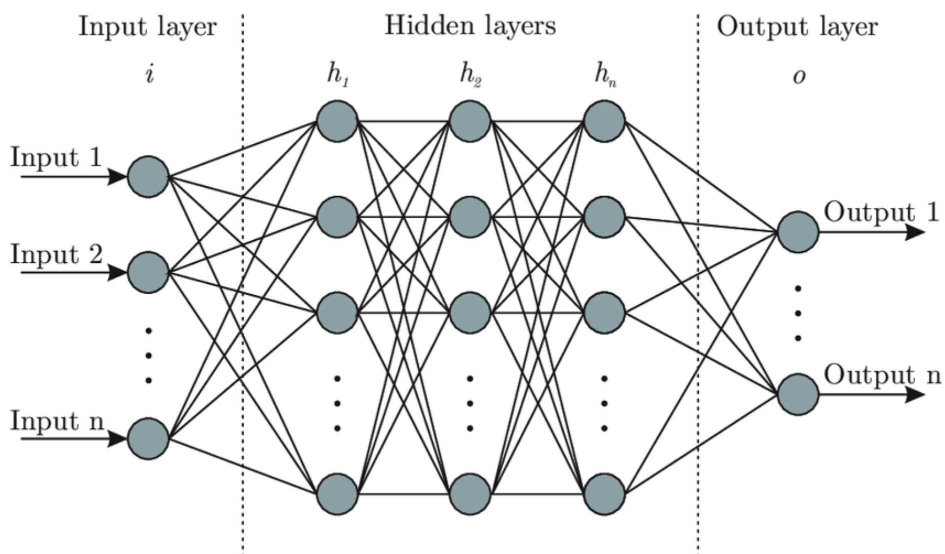
Neural Networks

Introduction

Neural Networks are machine-learning models inspired by the human brain. They consist of layers of interconnected processing units, known as neurons, that transform input data through weighted connections and activation functions. Neural networks are widely used for tasks like classification, regression, and feature extraction, forming the backbone of deep learning.



Architecture of Neural Networks

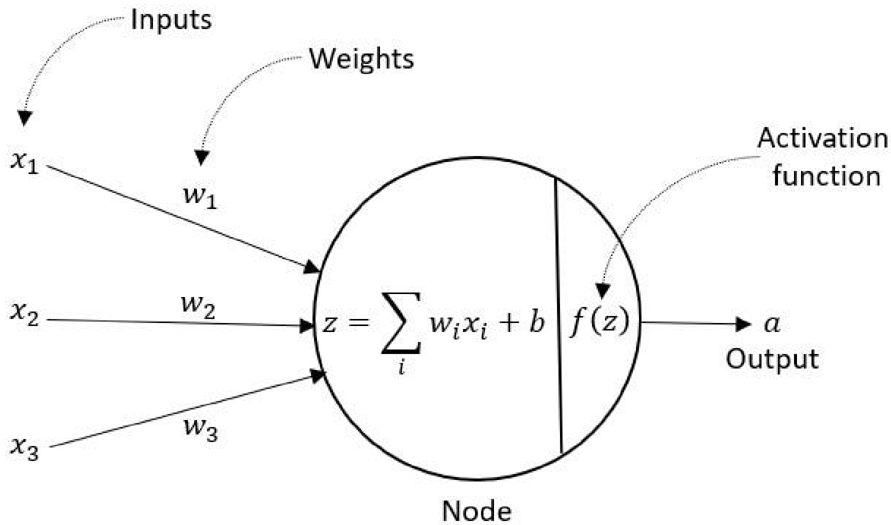
A Neural Network consists of multiple layers:

- Input Layer: Receives raw input features.
- Hidden Layers: Transforms inputs through weighted connections and non-linear activation functions.
- Output Layer: Produces the final prediction.

A fully connected L-layer neural network can be mathematically expressed as:

$$z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)} \tag{1}$$

$$a^{(l)} = f(z^{(l)}) \tag{2}$$



Where:

- $W^{(l)}$ and $b^{(l)}$ are the weight matrix and bias vector for layer l.
- $a^{(l)}$ is the activation for layer l.
- f is a non-linear activation function.

Activation Functions

Neural networks rely on non-linear activation functions:

- Sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$ (Used for binary classification).
- ReLU (Rectified Linear Unit): $f(z) = \max(0, z)$ (Common in deep networks).
- Tanh: $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ (Centers data around zero).
- Softmax: $f(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$ (Used for multi-class classification).

Learning in Neural Networks

1. Forward Propagation

For each input x, activations propagate through the network until the final output $\hat{y}$:

$$a^{(L)} = f(z^{(L)}) \tag{3}$$

2. Loss Function

The loss function measures how well the model's predictions match the true values:

- Mean Squared Error: Used for regression.
- Cross-Entropy Loss: Used for classification.

3. Backpropagation

To minimize the loss, gradients are computed and propagated backward:

$$\frac{\partial C}{\partial W^{(l)}} = \delta^{(l)} \cdot (a^{(l-1)})^T \tag{4}$$

$$\delta^{(l)} = (W^{(l+1)})^T \delta^{(l+1)} \odot f'(z^{(l)}) \tag{5}$$

4. Gradient Descent Optimization

Weights and biases are updated as follows:

$$W^{(l)} \leftarrow W^{(l)} - \alpha \frac{\partial C}{\partial W^{(l)}} \tag{6}$$

$$b^{(l)} \leftarrow b^{(l)} - \alpha \frac{\partial C}{\partial b^{(l)}} \tag{7}$$

where α is the learning rate.

Types of Neural Networks

- Feedforward Neural Networks: Standard dense networks.
- Convolutional Neural Networks: Used for image processing.
- Recurrent Neural Networks: Designed for sequential data.
- Deep Neural Networks: Networks with multiple hidden layers.
- Generative Adversarial Networks: Used for generative modeling.

Strengths

- Non-Linear Modeling: Captures complex relationships.
- Feature Learning: Automatically learns relevant representations.
- Scalability: Efficient for large datasets.
- Versatility: Applied in images, text, and structured data.

Weaknesses

- Computational Complexity: Requires significant resources.
- Data Requirements: Needs large amounts of labeled data.
- Lack of Interpretability: Hard to understand model decisions.
- Overfitting: Sensitive to excessive parameter tuning.

Applications

- Image Recognition: Object detection and classification.
- Natural Language Processing: Sentiment analysis, machine translation.
- Speech Recognition: Converting audio to text.
- Autonomous Vehicles: Processing sensor data for self-driving cars.
- Medical Diagnosis: Identifying diseases from medical images.
- Recommendation Systems: Personalizing content suggestions.

Conclusion

Neural Networks have transformed machine learning and AI, enabling breakthroughs in diverse applications. While they come with computational challenges, their ability to model complex data makes them a cornerstone of modern deep-learning techniques.