Concept and Operation Principles of Artificial Neural Networks

Artificial Neural Network (ANN) is a computational model inspired by the biological nervous system, designed to simulate and process complex information tasks.

Artificial Neural Networks are composed of numerous neurons (also known as nodes), and these neurons are interconnected through connections (referred to as weights), forming a network structure that simulates the interactions between nerve cells in the brain.

Components and Operational Principles of Artificial Neural Networks (ANN):

Neurons: Neurons are the fundamental units of a neural network; they receive inputs, perform computations, and generate outputs. Each neuron has an activation function that converts input signals into output signals. Typical activation functions include Sigmoid, ReLU (Rectified Linear Unit), and Tanh (Hyperbolic Tangent), among others.

Layers: Neural networks typically consist of multiple layers, including input, hidden, and output layers. The input layer receives raw data, the hidden layers extract features, and the output layer generates the final predictions or results.

Connection Weights: Each connection between neurons has a weight that represents the strength of information transmission between different neurons. These weights are automatically learned during the training process to enable the neural network to adapt to specific tasks.

Feedforward: In the feedforward process, input signals pass from the input layer to the output layer through connections between neurons, ultimately producing predictions or outputs. The forward propagation is achieved by computing the weighted sum of each neuron and applying the activation function.

Training: The training process of a neural network is accomplished through the backpropagation algorithm. During training, the network compares predictions with labeled data, calculates the error between predictions and actual values, and reduces the error by adjusting connection weights. This process is implemented through gradient descent to minimize the loss function.

Loss Function: The loss function measures the difference between the model's predictions and the actual values. The objective of training is to minimize the value of the loss function.

Activation Function: After computing the weighted sum of inputs within a neuron, the activation function transforms the result into the output of the neuron. They introduce non-linearity, enabling the neural network to capture more complex patterns.

Backpropagation: Backpropagation is an iterative optimization process used to adjust connection weights based on the gradient of the loss function to enhance the performance of the neural network. This process involves propagating error signals backward from the output layer and updating weights accordingly.

Deep Neural Networks: Neural networks that encompass multiple hidden layers are termed deep neural networks. They perform exceptionally well in handling complex problems and large-scale datasets, such as image recognition, natural language processing, and speech recognition.

Application Areas: Artificial neural networks are applied in nearly every field of machine learning and artificial intelligence, including image recognition, speech recognition, natural language processing, recommendation systems, autonomous driving, financial forecasting, and more.

Next, let's delve into the following 6 sections concerning ANN:

Feedforward Neural Network

The feedforward neural network is inspired by the neurons of the human brain. Its operation involves the forward transmission of information between different layers, akin to a conveyor belt. This transmission is unidirectional and doesn't form loops, hence the term "feedforward."

Basic Principles:

A feedforward neural network comprises multiple neurons organized into various layers: input, hidden, and output.

Input Layer: This layer receives the data you provide, such as pixel values in images or words in text.

Hidden Layer: It serves as the core of the network, responsible for processing input data. It may consist of multiple layers, each conducting various mathematical operations.

Output Layer: This layer provides the final result, such as object types in images or sentiment in text.

Formula Explanation:

Neurons serve as the fundamental building blocks of the feedforward neural network.

A neuron receives a set of inputs, aggregates them, and then passes the result through a function to generate an output.

The feedforward neural network serves as the cornerstone of deep learning, simulating the operation of neurons in the brain. These networks adapt to various tasks by learning weights and biases.

Convolutional Neural Network (CNN)

Imagine recognizing a dog in an image; you'd notice local features like eyes, nose, ears, and then combine these features to identify it as a dog. CNN simulates this process in a machine.

Basic Principles:

The core idea of a Convolutional Neural Network lies in convolution operations. Convolution is a mathematical operation that detects features in an input image by sliding a small window (commonly called a kernel or filter) over the image.

This kernel continually moves across the image, calculating a weighted sum for each local region, generating a feature map. Each element in this feature map represents the intensity of the detected feature.

Convolution operations possess a local nature, focusing on a small portion of the image, similar to how humans observe images. This property grants CNNs a level of invariance to translation, rotation, and scaling, as they can detect the same features regardless of their positions in the image.

CNNs extract features from images using convolution operations, enabling tasks like image classification.

Recurrent Neural Network (RNN)

RNN acts as a model with memory, capable of processing sequential data such as text, audio, or time-series. By continually passing information and maintaining internal states, RNN can understand the context of the data.

Basic Principles:

The fundamental building block of an RNN is the neuron, which accepts input and internal state and produces an output. The internal state serves as the network's memory, storing previously seen information.

RNN includes a recurrent connection allowing information to transfer between different time steps, resembling the pages of a book. You can move from one page to another, enabling RNN to handle sequences of varying lengths.

LSTM functions much like a person with memory, capable of remembering essential information and discarding unimportant details. It excels at handling long sequential data, as it captures and retains critical information over extended sequences without getting overwhelmed by irrelevant data.

Basic Principles:

The core idea behind LSTM involves the concepts of the cell state and gates. The cell state functions as the memory of the LSTM, capable of transmitting information and retaining or discarding it when necessary.

Gates are used to control the flow of information, facilitating the forgetting of unnecessary information and remembering important details.

The working process of LSTM involves three primary steps: forget, store, and update.

Forget: The cell state determines which information should be forgotten and what should be retained. Gates control the forgetting process.

Store: New information is added to the cell state to update the memory. Gates also control the information storage process.

Update: Based on the current input and cell state, LSTM generates new output and cell state, which becomes the input for the next time step. This process involves updating the information according to the current context.

LSTM's capability to manage and retain information over extended sequences has made it significantly effective in tasks involving long-term dependencies, such as language modeling, speech recognition, and time-series predictions.

(Self-Attention Model) - Transformer

The Transformer is a neural network model capable of understanding text and sequential data. Its unique feature lies in employing self-attention mechanisms, allowing it to simultaneously focus on different parts of the input data, unlike traditional recurrent neural networks (RNN) or convolutional neural networks (CNN), which depend on fixed window sizes or sequence orders.

Basic Principles:

The core idea of the Transformer involves segmenting input data into distinct "word embeddings" and then employing self-attention mechanisms to determine the relationships between these word embeddings. This approach enables the model to handle long texts and capture intricate relationships between different words.

Self-Attention Mechanism:

The self-attention mechanism is fundamental to the Transformer. It computes the importance of each position in the input sequence concerning other positions. This importance is determined by calculating a weight based on the similarity of the input. Crucially, this calculation is completed based on the input data itself, independent of sequence length.

Specific Steps:

Embedding Layer: Transforms the input text sequence into vector form, where each word corresponds to a vector trained to possess semantic information.

Self-Attention Computation: For each word, computes its similarity score with all other words and uses these scores as weights to weight other word embedding vectors. This process enables the model to focus more on words related to the current word.

Multi-Head Self-Attention: Transformer employs multiple self-attention heads to capture different levels of relationships, each generating a set of weights that are ultimately merged.

Residual Connection and Layer Normalization: Adds the output of multi-head self-attention to the input and applies layer normalization to prevent gradient vanishing or explosion.

Feed-Forward Network: Applies non-linear transformations to vectors at each position to enhance the model's representational capabilities.

Encoder and Decoder: In tasks like machine translation, the Transformer typically consists of an encoder for handling input and a decoder for generating output.

(GAN)

The core idea behind Generative Adversarial Networks (GANs) is to simulate the way humans create things.

GANs consist of two primary components: the Generator and the Discriminator. These two parts engage in a game, where the Generator gradually learns to create realistic data, while the Discriminator becomes better at distinguishing between real and fake data.

Basic Principles:

Generator: The Generator's role is to take in a random noise vector and transform it into realistic data, such as images. It's a neural network that adjusts its parameters continually to generate data that closely resembles real data.

Discriminator: The Discriminator aims to differentiate data generated by the Generator from real data. It's also a neural network that evaluates input data, outputting a probability value between 0 and 1 to indicate the authenticity of the data.

Adversarial Training: The Generator and Discriminator alternate during training. The Generator strives to produce more realistic data, while the Discriminator aims to better differentiate between real and fake data. This adversarial process drives the Generator to continuously enhance the quality of the generated data.

Generative Adversarial Networks find applications in various creative fields. They rely on the adversarial interplay between the Generator and Discriminator to continuously optimize the Generator to create more realistic data.