

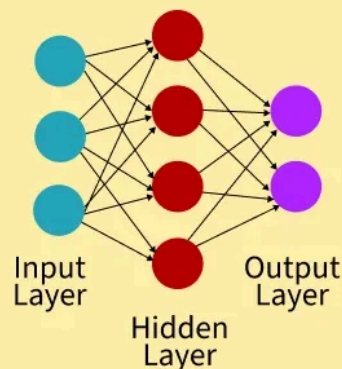
# Artificial Neural Networks and its Applications

Last Updated : 12 Jul, 2025

Artificial Neural Networks (ANNs) are computer systems designed to mimic how the human brain processes information. Just like the brain uses neurons to process data and make decisions, ANNs use artificial neurons to analyze data, identify patterns and make predictions. These networks consist of layers of interconnected neurons that work together to solve complex problems. The key idea is that ANNs can "learn" from the data they process, just as our brain learns from experience. They are used in various applications from recognizing images to making personalized recommendations. In this article, we will see more about ANNs, how they function and other core concepts.

## What Are Artificial Neural Networks?

- Simulate the human brain's information processing.
- Consist of input, hidden and output layers.
- Learn patterns and make predictions from data.

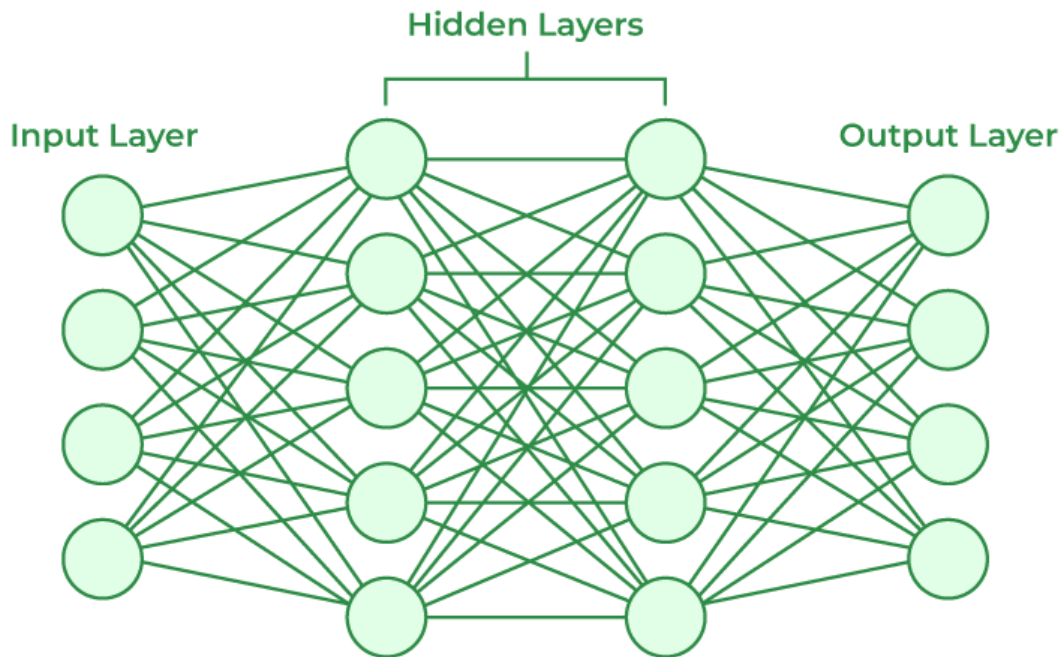


1 / 4

## Key Components of an ANN

- **Input Layer:** This is where the network receives information. For example, in an image recognition task, the input could be an image.
- **Hidden Layers:** These layers process the data received from the input layer. The more hidden layers there are, the more complex patterns the network can learn and understand. Each hidden layer transforms the data into more abstract information.

- **Output Layer:** This is where the final decision or prediction is made. For example, after processing an image, the output layer might decide whether it's a cat or a dog.



*Neural Networks Architecture*

## Working of Artificial Neural Networks

ANNs work by learning patterns in data through a process called training. During training, the network adjusts itself to improve its accuracy by comparing its predictions with the actual results.

Lets see how the learning process works:

- **Input Layer:** Data such as an image, text or number is fed into the network through the input layer.
- **Hidden Layers:** Each neuron in the hidden layers performs some calculation on the input, passing the result to the next layer. The data is transformed and abstracted at each layer.
- **Output Layer:** After passing through all the layers, the network gives its final prediction like classifying an image as a cat or a dog.

The process of [backpropagation](#) is used to adjust the weights between neurons. When the network makes a mistake, the weights are updated to reduce the error and improve the next prediction.

## Training and Testing:

- During training, the network is shown examples like images of cats and learns to recognize patterns in them.
- After training, the network is tested on new data to check its performance. The better the network is trained, the more accurately it will predict new data.

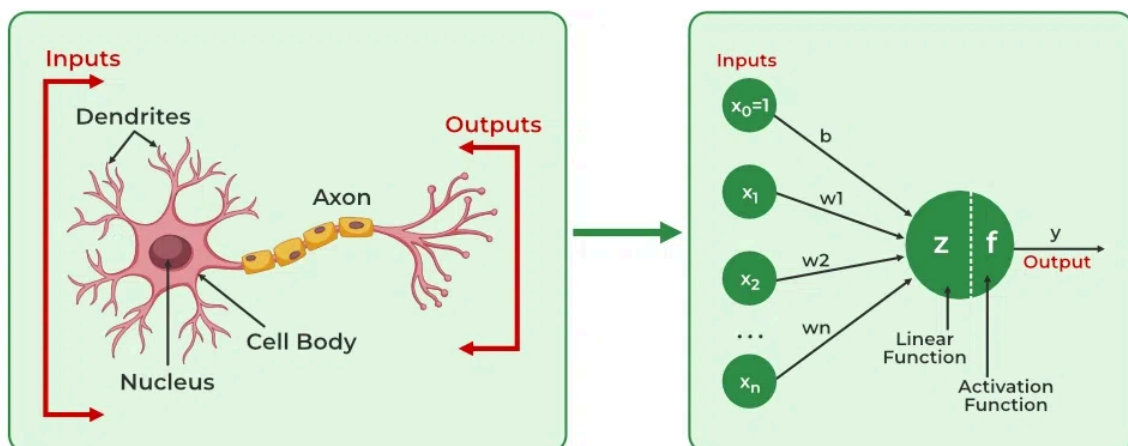
## How do Artificial Neural Networks learn?

- Artificial Neural Networks (ANNs) learn by training on a set of data. For example, to teach an ANN to recognize a cat, we show it thousands of images of cats. The network processes these images and learns to identify the features that define a cat.
- Once the network has been trained, we test it by providing new images to see if it can correctly identify cats. The network's prediction is then compared to the actual label (whether it's a cat or not). If it makes an incorrect prediction, the network adjusts by fine-tuning the weights of the connections between neurons using a process called backpropagation. This involves correcting the weights based on the difference between the predicted and actual result.
- This process repeats until the network can accurately recognize a cat in an image with minimal error. Essentially, through constant training and feedback, the network becomes better at identifying patterns and making predictions.

## Artificial neurons vs Biological neurons

Aspect	Biological Neurons	Artificial Neurons
Structure	Dendrites: Receive signals from other neurons.	Input Nodes: Receive data and pass it on to the next layer.

Aspect	Biological Neurons	Artificial Neurons
	<b>Cell Body (Soma):</b> Processes the signals.	<b>Hidden Layer Nodes:</b> Process and transform the data.
	<b>Axon:</b> Transmits processed signals to other neurons.	<b>Output Nodes:</b> Produce the final result after processing.
<b>Connections</b>	<b>Synapses:</b> Links between neurons that transmit signals.	<b>Weights:</b> Connections between neurons that control the influence of one neuron on another.
<b>Learning Mechanism</b>	<b>Synaptic Plasticity:</b> Changes in synaptic strength based on activity over time.	<b>Backpropagation:</b> Adjusts the weights based on errors in predictions to improve future performance.
<b>Activation</b>	<b>Activation:</b> Neurons fire when signals are strong enough to reach a threshold.	<b>Activation Function:</b> Maps input to output, deciding if the neuron should fire based on the processed data.



## Common Activation Functions in ANNs

Activation functions are important in neural networks because they introduce non-linearity and helps the network to learn complex patterns. Lets see some common activation functions used in ANNs:

1. **Sigmoid Function**: Outputs values between 0 and 1. It is used in binary classification tasks like deciding if an image is a cat or not.
2. **ReLU (Rectified Linear Unit)**: A popular choice for hidden layers, it returns the input if positive and zero otherwise. It helps to solve the [vanishing\\_gradient problem](#).
3. **Tanh (Hyperbolic Tangent)**: Similar to sigmoid but outputs values between -1 and 1. It is used in hidden layers when a broader range of outputs is needed.
4. **Softmax**: Converts raw outputs into probabilities used in the final layer of a network for multi-class classification tasks.
5. **Leaky ReLU**: A variant of ReLU that allows small negative values for inputs helps in preventing “dead neurons” during training.

These functions help the network decide whether to activate a neuron helps it to recognize patterns and make predictions.

*For more details refer to [Types of Activation Functions](#)*

## Types of Artificial Neural Networks

### 1. Feedforward Neural Network (FNN)

[Feedforward Neural Networks](#) are one of the simplest types of ANNs. In this network, data flows in one direction from the input layer to the output layer, passing through one or more hidden layers. There are no loops or cycles means the data doesn't return to any earlier layers. This type of network does not use backpropagation and is mainly used for basic classification and regression tasks.

### 2. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are designed to process data that has a grid-like structure such as images. It include convolutional layers that apply filters to extract important features from the data such as edges or textures. This makes CNNs effective in image and speech recognition as they can identify patterns and structures in complex data.

### 3. Radial Basis Function Network (RBFN)

Radial Basis Function Networks are designed to work with data that can be modeled in a radial or circular way. These networks consist of two layers: one that maps input to radial basis functions and another that finds the output. They are used for classification and regression tasks especially when the data represents an underlying pattern or trend.

### 4. Recurrent Neural Network (RNN)

Recurrent Neural Networks are designed to handle sequential data such as time-series or text. Unlike other networks, RNNs have feedback loops that allow information to be passed back into previous layers, giving the network memory. This feature helps RNNs to make predictions based on the context provided by previous data helps in making them ideal for tasks like speech recognition, language modeling and forecasting.

## Optimization Algorithms in ANN Training

Optimization algorithms adjust the weights of a neural network during training to minimize errors. The goal is to make the network's predictions more accurate. Lets see key algorithms:

1. **Gradient Descent:** Most basic optimization algorithm that updates weights by calculating the gradient of the loss function.
2. **Adam (Adaptive Moment Estimation):** An efficient version of gradient descent that adapts learning rates for each weight used in deep learning.



3. **RMSprop:** A variation of gradient descent that adjusts the learning rate based on the average of recent gradients, it is useful in training recurrent neural networks (RNNs).
4. **Stochastic Gradient Descent (SGD):** Updates weights using one sample at a time helps in making it faster but more noisy.

*For more details refer to [Optimization Algorithms in ANN](#)*

## Applications of Artificial Neural Networks

1. **Social Media:** ANNs help social media platforms suggest friends and relevant content by analyzing user profiles, interests and interactions. They also assist in targeted advertising which ensures users to see ads tailored to their preferences.
2. **Marketing and Sales:** E-commerce sites like Amazon use ANNs to recommend products based on browsing history. They also personalize offers, predict customer behavior and segment customers for more effective marketing campaigns.
3. **Healthcare:** ANNs are used in medical imaging for detecting diseases like cancer and they assist in diagnosing conditions with accuracy similar to doctors. Additionally, they predict health risks and recommend personalized treatment plans.
4. **Personal Assistants:** Virtual assistants like Siri and Alexa use ANNs to process natural language, understand voice commands and respond accordingly. They help manage tasks like setting reminders helps in making calls and answering queries.
5. **Customer Support:** ANNs power chatbots and automated customer service systems that analyze customer queries and provide accurate responses helps in improving efficiency in handling customer inquiries.
6. **Finance:** In the financial industry, they are used for fraud detection, credit scoring and predicting market trends by analyzing large sets of transaction data and spotting anomalies.

## Challenges in Artificial Neural Networks

1. **Data Dependency:** ANNs require large amounts of high-quality data to train effectively. Gathering and cleaning sufficient data can be time-consuming, expensive and often impractical especially in industries with limited access to quality data.
2. **Computational Power:** Training deep neural networks with many layers, demands significant computational resources. High-performance hardware (e.g GPUs) is often required which makes it expensive and resource-intensive.
3. **Overfitting:** It can easily overfit to the training data which means they perform well on the training set but poorly on new, unseen data. This challenge arises when the model learns to memorize rather than generalize, reducing its real-world applicability.
4. **Interpretability:** They are often referred to as "black boxes." It is difficult to understand how they make decisions which is a problem in fields like healthcare and finance where explainability and transparency are important.
5. **Training Time:** Training ANNs can take a long time, especially for deep learning models with many layers and vast datasets. This lengthy training process can delay the deployment of models and hinder their use in time-sensitive applications.

As technology keeps improving, Artificial Neural Networks will continue to change the way we solve problems and make our lives easier.



## Structure and Components of Artificial Neural Networks (ANN)

[Visit Course](#)

[Comment](#)

[More info](#)

## Explore

[Introduction to AI](#)

---

[AI Concepts](#)

---

[Machine Learning in AI](#)

---

[Robotics and AI](#)

---

[Generative AI](#)

---

[AI Practice](#)

---



#### Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

#### Registered Address:

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305



### Company

About Us  
Legal  
Privacy Policy  
Contact Us  
Advertise with us  
GFG Corporate Solution  
Campus Training Program

### Explore

POTD  
Job-A-Thon  
Community  
Blogs  
Nation Skill Up

### Tutorials

Programming Languages  
DSA  
Web Technology  
AI, ML & Data Science  
DevOps  
CS Core Subjects  
Interview Preparation  
GATE  
Software and Tools

### Courses

IBM Certification  
DSA and Placements  
Web Development  
Programming Languages  
DevOps & Cloud  
GATE  
Trending Technologies

### Videos

DSA  
Python  
Java  
C++  
Web Development  
Data Science  
CS Subjects

### Preparation Corner

Aptitude  
Puzzles  
GfG 160  
DSA 360  
System Design