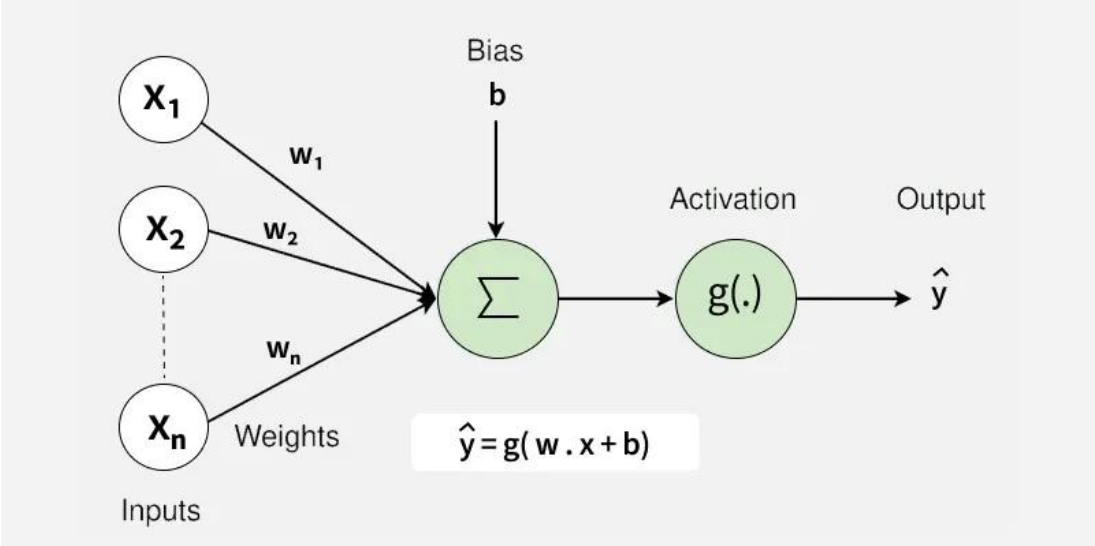
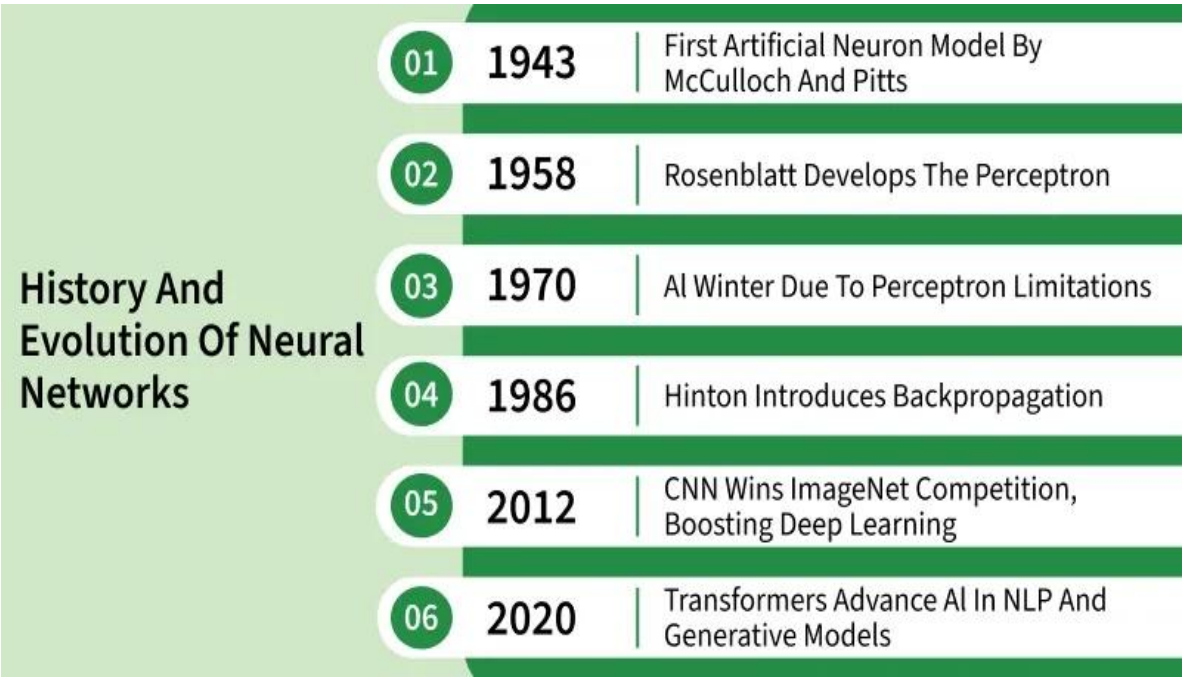
NEURAL NETWORKS BASICS

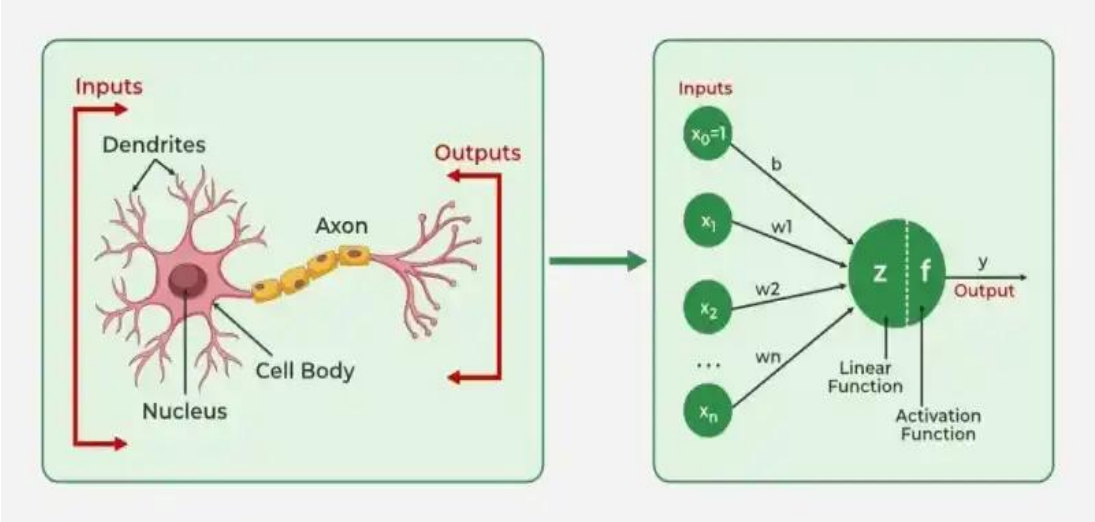
- Neural Networks are machine learning models inspired by the way the human brain processes information.
- They consist of interconnected layers of neurons that learn patterns from data and make predictions.
- Learn complex patterns from large datasets.
- Consist of input, hidden, and output layers.
- Used in image recognition, text processing, speech recognition, and prediction tasks.





key components

- Neural Networks are built from several key components:
- **Neurons:** The basic units that receive inputs, each neuron is governed by a threshold and an activation function.
- **Connections:** Links between neurons that carry information, regulated by weights and biases.
- **Weights and Biases**: These parameters determine the strength and influence of connections.
- **Propagation Functions**: Mechanisms that help process and transfer data across layers of neurons.
- **Learning Rule**: The method that adjusts weights and biases over time to improve accuracy.
- Learning in neural networks follows a structured, three-stage process:
- **Input Computation**: Data is fed into the network.
- **Output Generation**: Based on the current parameters, the network generates an output.
- **Iterative Refinement**: The network refines its output by adjusting weights and biases, gradually improving its performance on diverse tasks.

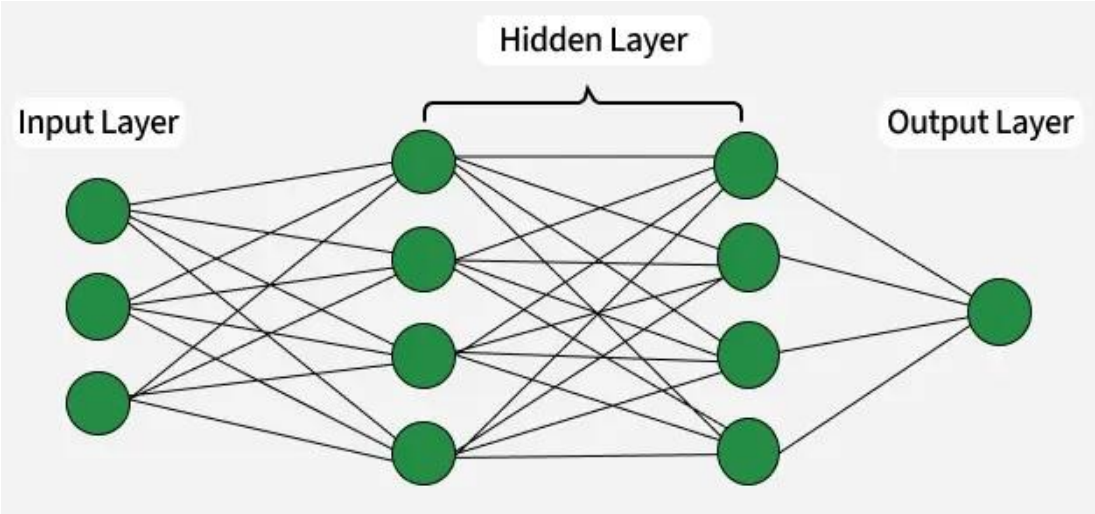


Importance of Neural Networks

- Identify Complex Patterns: Recognize intricate structures and relationships in data; adapt to dynamic and changing environments.
- Learn from Data: Handle vast datasets efficiently; improve performance with experience and retraining.
- Drive Key Technologies: Power natural language processing (NLP); enable self-driving vehicles; support automated decision-making systems.
- Boost Efficiency: Streamline workflows and processes; enhance productivity across industries.

- Backbone of AI: Serve as the core driver of artificial intelligence progress; continue shaping the future of technology and innovation.

Layers in Neural Network Architecture



- **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.
- **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.
- **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. <u>Forward Propagation</u>

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:

$$z=w_1x_1+w_2x_2+\ldots+w_nx_n+b$$

where
W represents the weights
X represents the inputs
B is the bias

2. Activation: The result of the linear transformation (denoted as 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.

EXAMPLE OF EMAIL CLASSIFICATION

- Let's consider a record of an email dataset:

Email ID	Email Content	Sender	Subject Line	Label
1	"Get free gift cards now!"	spam@example.com	"Exclusive Offer"	1

To classify this email, we will create a feature vector based on the analysis of keywords such as "free" "win" and "offer"

- The resulting feature vector is:
- free: Present (1)
- win: Absent (0)
- offer: Present (1)

Feature Vector:

- [1, 0, 1]
- The feature vector is passed through the neural network, where each neuron computes a weighted sum of the inputs and applies an activation function to produce its output.

Hidden Layer Calculation:

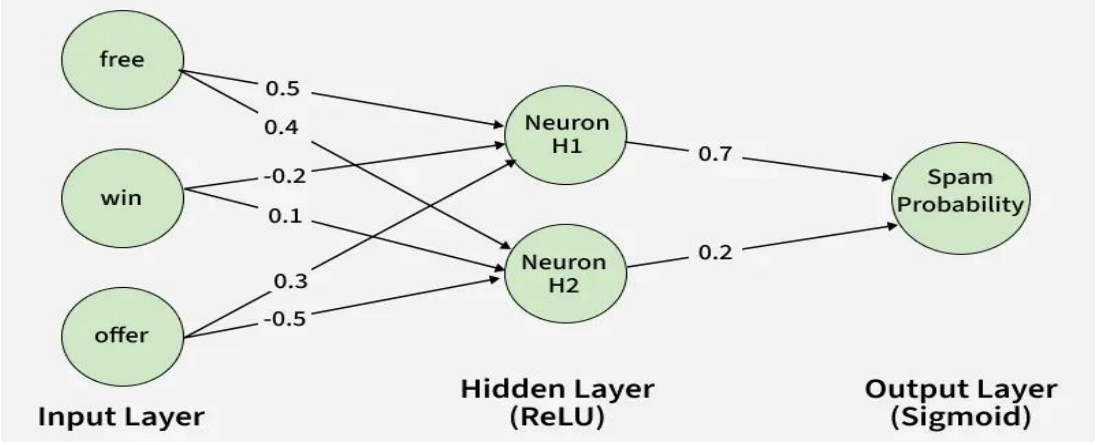
- Assume the hidden layer contains two neurons with the following weights
- Neuron H1: [0.5, −0.2, 0.3]
- Neuron H2: [0.4, 0.1, −0.5]
- For the input vector [1, 0, 1], the weighted sums are:
- For H1: (1 × 0.5) +(0 ×− 0.2) + (1 × 0.3) = 0.5 + 0+ 0.3 = 0.8
- For H2: (1 × 0.4) +(0 × 0.1) + (1 × −0.5) = 0.4 + 0 −0.5 =−0.1
- Applying the ReLU activation function:
- H1 Output: ReLU(0.8) = 0.8
- H2 Output: ReLu(-0.1) = 0

Output Layer

Output Layer: The outputs from the hidden layer are then passed to the output neuron.
Output Weights: [0.7, 0.2]
Input from Hidden Layer: [0.8, 0]
Weighted Sum: (0.8×0.7)+(0×0.2)=0.56+0=0.56
Activation (Sigmoid): σ(0.56)=1/1+e^−0.56≈0.636

Final Classification:

- The output value of approximately 0.636 indicates the probability of the email being spam.
- Since this value is greater than 0.5, the neural network classifies the email as spam (1).
-



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.

Applications

- Image and Video Recognition: CNNs are extensively used in applications such as facial recognition, autonomous driving and medical image analysis.
- Natural Language Processing (NLP): RNNs and transformers power language translation, chatbots and sentiment analysis.
- Finance: Predicting stock prices, fraud detection and risk management.
- Healthcare: Neural networks assist in diagnosing diseases, analyzing medical images and personalizing treatment plans.
- Gaming and Autonomous Systems: Neural networks enable real-time decision-making, enhancing user experience in video games and enabling autonomous systems like self-driving cars.

Advantages

- Neural networks use non-linear activation functions to learn intricate patterns that cannot be captured by linear models.
- They automatically extract relevant features from raw data, reducing the need for manual feature engineering.
- Multiple neurons can perform computations simultaneously, improving computational efficiency, especially on GPUs.
- Once trained, neural networks can make accurate predictions on new data that was not part of the training set.

Limitations

- Training large neural networks is computationally intensive and requires significant processing power.
- Act as black box models, making it difficult to understand how decisions are made, which can be a concern in critical applications.
- May suffer from overfitting, where the model memorizes training data instead of learning general patterns, even though regularization can help reduce this issue.
- Neural networks often require large, well-labeled datasets for effective training and performance may suffer if the data is limited or biased.