Deep learning

Deep learning is a type of machine learning that uses artificial neural networks to learn from data. These networks have **layers of "neurons"** that process information step by step. The more layers, the **deeper** the learning. Artificial neural networks are inspired by the human brain, and they can be used to solve a wide variety of problems, including image recognition, natural language processing, and speech recognition.

Why Deep Learning?

Deep learning is powerful because it can learn patterns from a huge amount of data and make smart decisions without human instructions. It helps in:

- Recognizing images (Face recognition, medical scans)
- Understanding speech (Voice assistants like Siri, Alexa)
- Translating languages (Google Translate)
- Self-driving cars (Detecting objects, making driving decisions)
- Chatbots & AI tools (Like ChatGPT)

Why Now?

Deep learning has been around for years, but now it's growing fast because:

More powerful computers – Faster GPUs and TPUs help train big models.

More data available – The internet, social media, and sensors provide massive amounts of data.

Better algorithms – New techniques make deep learning more efficient and accurate.

Cloud computing – AI models can be trained online without needing expensive hardware.

What is a Neural Network?

A **Neural Network** is a type of artificial intelligence (AI) model inspired by the human brain. It is used in **machine learning** and **deep learning** to recognize patterns and make predictions.

A neural network consists of **layers of interconnected neurons** that process data step by step.

Perceptron

A Perceptron is the simplest type of neural network used in deep learning. It is like a small brain cell that makes decisions based on inputs.

It is used for simple decision-making, like recognizing handwritten digits or classifying emails as spam or not spam.

Example:

If you train a perceptron to recognize cats, it checks features (like ears, whiskers, and tail) and decides whether the image is a cat or not!

Perceptron: Forward Propagation

Forward propagation is the process of passing inputs through a perceptron to get an output.

Steps of Forward Propagation:

input Layer:

- The perceptron takes multiple inputs (e.g., numbers, pixel values).
- Example: x_1, x_2, x_3 (features of an image).

Weights & Bias:

- Each input has a weight (w_1, w_2, w_3) that decides its importance.
- A bias (b) helps shift the output to improve accuracy.

Summation Function:

- The perceptron adds up all inputs multiplied by their weights and adds the bias:

$$Z = (x_1 \times w_1) + (x_2 \times w_2) + (x_3 \times w_3) + b$$

Activation Function:

- The sum Z is passed through an activation function to decide the output.
- Example: The Step Function outputs 1 if Z is above a threshold and 0 otherwise.

Output:

- The perceptron gives a final decision (e.g., "Yes" or "No", "Spam" or "Not Spam").

This process is called forward propagation because data moves forward through the perceptron to get a result! 🚀

What is an Activation Function?

An **activation function** is a mathematical function used in a neural network to **decide the output** of a neuron. It helps the network **learn complex patterns** by introducing non-linearity.

Without activation functions, a neural network would behave like a simple equation and wouldn't be able to **learn complex data**, like images or speech.

Types of Activation Functions

Sigmoid Function

- Formula: f(x)=1/1+e*-x
- Output: Between **0 and 1** (good for probability tasks)
- Limitation: Can slow down training (vanishing gradient problem)

ReLU (Rectified Linear Unit

- Formula: f(x)=max⁡(0,x)f(x) = \max(0, x)
- Output: **0 for negative values**, same as input for positive values
- Used in **deep learning** (faster and better performance)

Tanh (Hyperbolic Tangent 📈)

- Formula: f(x)=ex−e−xex+e−xf(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}
- Output: Between **-1 and 1** (better than Sigmoid

What is a Multi-Output Perceptron?

A **multi-output perceptron** is a perceptron that has **multiple neurons in the output layer**, meaning it can produce **more than one output** at the same time.

Example:

🔹 ***Weather Prediction:*** *A model can predict both temperature and humidity at the same time.*

What is a Dense Layer?

A **dense layer** is a layer in a neural network where **each neuron is connected to every neuron in the previous layer**.

Quantifying Loss:

The loss of our network measures the cost incurred from incorrect predictions.

Empiricl Loss:

The empirical loss measures total loss over our entire dataset.

Computing Gradients with Backpropagation

Backpropagation works by **computing the gradient of the loss function** with respect to each weight in the network, moving backward from the output layer to the input layer.

Using the **chain rule**, we compute how much each weight contributes to the error and adjust accordingly.

Regularization in Deep Learning

Regularization is a technique used to **prevent overfitting** in machine learning models. It helps the model generalize better to new data by reducing its dependence on specific training examples.

- When a model is **too complex**, it memorizes the training data instead of learning general patterns.
- This leads to **overfitting**, meaning the model performs well on training data but poorly on new data.
- Regularization helps **simplify the model** and improve its performance on unseen data

Dropout

- **Randomly deactivates** neurons during training to prevent over-reliance on specific neurons.
- Helps **spread learning across the network**, improving generalization.

Early Stopping

- Stops training when validation loss **stops improving**, preventing overfitting.

What is Batching in Deep Learning?

Batching is a technique used to **process multiple data samples together** instead of one at a time. It helps speed up training and improves model performance.

What is Deep Sequence Modeling?

Deep Sequence Modeling is a deep learning technique used to process and analyze **sequential data**, where the order of data points matters. It helps models **learn patterns over time**, making it useful for tasks like speech recognition, language translation, and time-series forecasting.

In many real-world problems, data comes in sequences, such as:
✓ **Text data** → Sentences have word order
✓ **Speech & audio** → Sounds change over time (
✓ **Stock prices** → Financial trends depend on past data.

Applications:

- **Voice Recognition** – AI can understand and process speech (e.g., Google Assistant).
- **Music Generation** – AI composes new music by learning patterns in sound.

- **Emotion Detection** – Identifies emotions in speech (used in call centers).
- **stock Market Prediction** – Analyzing financial trends to predict stock prices.
- **Weather Forecasting** – Using historical weather data to predict future conditions.
- **Self-Driving Cars** – AI analyzes sensor data over time to make driving decisions.

What is a Recurrent Neural Network (RNN)?

A **Recurrent Neural Network (RNN)** is a type of artificial neural network designed to process **sequential data**, where the order of inputs matters. Unlike traditional neural networks, RNNs have a **memory** that allows them to remember previous inputs and use them for future predictions.

In a standard neural network, information flows **one way** (input → hidden layers → output). In an **RNN**, each neuron also **feeds back its output** to itself, allowing it to remember past information…

Backpropagation for Feedforward Models

Backpropagation is the core algorithm used to **train feedforward neural networks** by adjusting weights to minimize errors. It works by **propagating errors backward** from the output layer to the input layer using the **chain rule of calculu**

Self-Attention with Neural Networks

Self-Attention **is a mechanism in neural networks that allows models to** focus on important parts of an input sequence**, regardless of their position. Self-Attention assigns** different importance (weights) to different words or elements **in a sequence. A self-attention model** focuses on key words..

Goal:

Identify and attend to most important features in input

Steps:

- Encode Postion information
- Extract Query , key , value for search
- Compute attention weighting
- Extract features with high attention

What is a Loss Function?

A **loss function** is a mathematical function that measures how well a machine learning model's predictions match the actual values. It **quantifies the error** so the model can improve by adjusting its weights using **backpropagation**.

What is Gradient Descent?

Gradient Descent is an optimization algorithm used to **minimize the loss function** by adjusting the model's weights. It helps neural networks learn by gradually moving towards the best possible solution.

Encoding Language for Neural Networks

Neural networks cannot understand raw text, so we need to convert words into **numerical representations** before feeding them into a model. This process is called **language encoding**.

One-Hot Encoding (Basic)

- Represents each word as a **binary vector**.
- Example: If we have a vocabulary of 4 words:
  ["cat", "dog", "fish", "bird"], then:
  - "cat" → **[1, 0, 0, 0]**
  - "dog" → **[0, 1, 0, 0]**
  - "fish" → **[0, 0, 1, 0]**